با ایران توشه
یک قدم جلوتر
باش!

ایران توشه
توشه ای برای موفقیت



ایران توشه، مرجع دانلود فایل های کمک آموزشی دانش آموزان و کنکوری ها

همه چیز
برای موفقیت
تو!

- دانلود رایگان آزمون های آزمایشی قلمچی، خیلی سبز، گاج و ...
- دانلود رایگان گام به گام تمامی دروس
- دانلود رایگان نمونه سوالات امتحانی (نوبت اول و دوم) و امتحانات نهایی
- دانلود رایگان جزوه های کمک آموزشی و شب امتحانی

کافیست هر فایلی که نیاز دارید را در گوگل به این صورت جستجو کنید:

فایل مورد نیاز + ایران توشه

مثال:

دانلود گام به گام دوازدهم تجربی ایران توشه



irantooshe.ir

t.me/irantooshe

گام به گام توسعه برنامه‌سازی و پایگاه داده پایه یازدهم
پودمان اول

عنوان فصل: پیاده‌سازی پایگاه داده

از صفحه‌ی ۱ تا صفحه‌ی ۵۶ کتاب

در این فایل پاسخ فعالیت‌ها، کارهای کلاسی، تمرین‌ها و پرسش‌های فصل پیاده‌سازی پایگاه داده آمده است.
مباحث فصل: ایجاد پایگاه داده (موجودیت، صفت، ارتباط، نمودار ER، جدول و کلید)، توسعه‌ی پایگاه داده (درج، ویرایش، حذف، پرس‌وجو با SELECT و گزارش‌سازی)

تهیه‌شده در ایران‌توشه، توشه‌ای برای موفقیت

irantooshe.ir

فهرست پرسش‌ها

۱ صفحه ۳ · فعالیت گروهی — تاریخچه‌ی ذخیره‌سازی و بازیابی اطلاعات ۸

۲ صفحه ۴ · کنجکاوی — تعامل انسان با پایگاه داده ۸

۳ صفحه ۴ · فعالیت گروهی — داده‌های تکراری ۹

۴ صفحه ۵ · فعالیت گروهی — مشکلات جمع‌آوری پراکنده‌ی اطلاعات ۹

۵ صفحه ۶ · کنجکاوی — دفترچه‌ی تلفن بدون قاعده ۱۰

۶ صفحه ۷ · فعالیت گروهی — محیط‌های عملیاتی پیرامون ۱۰

۷ صفحه ۷ · کنجکاوی — میز و نیمکت به‌عنوان موجودیت ۱۰

۸ صفحه ۸ · پژوهش — تعریف موجودیت ۱۱

۹ صفحه ۸ · فعالیت کارگاهی — موجودیت‌های تاکسی تلفنی ۱۱

۱۰ صفحه ۸ · فعالیت کارگاهی — موجودیت‌های کتابخانه ۱۲

۱۱ صفحه ۸ · فعالیت کارگاهی — صفت‌های موجودیت ۱۲

ایران توشه
توشه‌ای برای موفقیت

۱۲ صفحه ۹ · کنجکاوی — معیار انتخاب صفت ۱۲
۱۳ صفحه ۹ · فعالیت گروهی — موجودیت، صفت و کلید ۱۳
۱۴ صفحه ۱۰ · پژوهش — معایب پایگاه داده ۱۳
۱۵ صفحه ۱۱ · فعالیت کارگاهی — ارتباط یک‌به‌یک ۱۴
۱۶ صفحه ۱۲ · فعالیت گروهی — ارتباط یک‌به‌چند ۱۴
۱۷ صفحه ۱۳ · کنجکاوی — نمایش گرافیکی رابطه‌ها ۱۵
۱۸ صفحه ۱۳ · فعالیت گروهی — ارتباط چندبه‌چند در کتابخانه ۱۵
۱۹ صفحه ۱۳ · فعالیت کارگاهی — نمودار ER فروشگاه و کافی‌نت ۱۶
۲۰ صفحه ۱۳ · فعالیت گروهی — نام ارتباط در مثال‌های چندبه‌چند ۱۷
۲۱ صفحه ۱۳ · فعالیت گروهی — تفاوت صفت کلیدی در نمودار ۱۷
۲۲ صفحه ۱۴ · فعالیت گروهی — تکمیل نمودار ER هنرآموز و هنرجو ۱۷
۲۳ صفحه ۱۴ · کنجکاوی — ارتباط موجودیت با خودش ۱۸
۲۴ صفحه ۱۶ · کنجکاوی — سیستم‌های مدیریت پایگاه داده ۱۹

ایران توشه
توشه‌ای برای موفقیت

۲۵ صفحه ۱۷ · فعالیت گروهی — موجودیت‌ها و فیلدهای هنرستان ۱۹

۲۶ صفحه ۱۸ · فعالیت کارگاهی — ایجاد پایگاه داده‌ی school ۱۹

۲۷ صفحه ۱۸ · پژوهش — قالب mdb و accdb ۲۰

۲۸ صفحه ۱۹ · کنجکاوی — ستون Click to Add ۲۰

۲۹ صفحه ۲۱ · فعالیت گروهی — انتخاب نوع داده‌ی فیلدها ۲۰

۳۰ صفحه ۲۲ · فعالیت کارگاهی — جدول Register ۲۱

۳۱ صفحه ۲۳ · فعالیت کارگاهی — جدول student در نمای Design ۲۲

۳۲ صفحه ۲۳ · کنجکاوی — روش‌های تعیین کلید اصلی ۲۳

۳۳ صفحه ۲۴ · کنجکاوی — پیام هشدار حذف فیلد ۲۳

۳۴ صفحه ۲۴ · فعالیت کارگاهی — کلید اصلی جدول student ۲۴

۳۵ صفحه ۲۵ · فعالیت کارگاهی — تغییر ساختار جدول student ۲۴

۳۶ صفحه ۲۵ · فعالیت کارگاهی — جدول‌های student، course، teacher و teach ۲۵

۳۷ صفحه ۲۸ · فعالیت کارگاهی — پایگاه داده‌ی Store ۲۷

ایران توشه
irantooshe.ir

۳۸ صفحه ۳۱ · کنجکاوی — روش‌های دیگر درج رکورد ۲۸

۳۹ صفحه ۳۴ · فعالیت کارگاهی — درج رکورد ۲۸

۴۰ صفحه ۳۴ · کارگاه ۲ — درج در جدول teach و کلید خارجی ۲۹

۴۱ صفحه ۳۵ · کنجکاوی — UPDATE بدون WHERE ۲۹

۴۲ صفحه ۳۵ · فعالیت کارگاهی — دستور UPDATE ۲۹

۴۳ صفحه ۳۶ · کنجکاوی — روش‌های دیگر حذف رکورد ۳۰

۴۴ صفحه ۳۷ · فعالیت کارگاهی — دستور DELETE ۳۰

۴۵ صفحه ۳۷ · کارگاه ۳ — حذف همه‌ی رکوردهای جدول ۳۱

۴۶ صفحه ۴۰ · فعالیت کارگاهی — پرس‌وجو با Wizard ۳۱

۴۷ صفحه ۴۲ · فعالیت کارگاهی — نام مستعار و ادغام فیلدها ۳۱

۴۸ صفحه ۴۳ · فعالیت کارگاهی — مرتب‌سازی نزولی ۳۲

۴۹ صفحه ۴۳ · کارگاه ۶ — آیا جدول اصلی مرتب می‌شود؟ ۳۲

۵۰ صفحه ۴۴ · فعالیت کارگاهی — مرتب‌سازی جدول course ۳۲

ایران توشه
توشه‌ای برای موفقیت

۵۱ صفحه ۴۴ · کارگاه ۷ — گروه‌بندی؛ تعیین فیلدها و جدول‌ها ۳۳

۵۲ صفحه ۴۵ · کارگاه ۷ — گروه‌بندی؛ شرط و WHERE ۳۳

۵۳ صفحه ۴۷ · فعالیت کارگاهی — گروه‌بندی بر اساس پایه و نام درس ۳۴

۵۴ صفحه ۵۰ · کارگاه ۸ — افزودن فیلد جدول دیگر به گزارش ۳۵

۵۵ صفحه ۵۱ · فعالیت کارگاهی — گزارش با Wizard ۳۵

۵۶ صفحه ۵۳ · فعالیت کارگاهی — ویرایش گزارش ثبت‌نام ۳۵

۵۷ صفحه ۵۴ · پروژه فاز ۱ — کتابخانه؛ موجودیت‌ها، صفات و نمودار ER ۳۶

۵۸ صفحه ۵۴ · پروژه فاز ۱ — کتابخانه؛ ایجاد پایگاه داده، جدول‌ها و ارتباط‌ها ۳۷

۵۹ صفحه ۵۴ · پروژه فاز ۱ — کتابخانه؛ ورود داده‌ها ۳۹

۶۰ صفحه ۵۴ · پروژه فاز ۱ — کتابخانه؛ پرس‌وجوهای ۱ تا ۴ ۳۹

۶۱ صفحه ۵۴ · پروژه فاز ۱ — کتابخانه؛ پرس‌وجوهای ۵ تا ۷ ۴۰

۶۲ صفحه ۵۴ · پروژه فاز ۱ — کتابخانه؛ گزارش‌ها ۴۲

ایران توشه
توشه‌ای برای موفقیت

۱ صفحه ۳ فعالیت گروهی

در خصوص موارد زیر در گروه گفت‌وگو کنید و نتیجه‌ی هر گروه را به کلاس ارائه دهید.

ـ نقش رایانه در ایجاد پایگاه‌های داده

ـ نگهداری اطلاعات به‌وسیله‌ی رایانه

ـ انفجار اطلاعات

ـ بشر از ابتدا تاکنون از چه ابزارهای ذخیره‌سازی استفاده کرده است؟

نقش رایانه در ایجاد پایگاه‌های داده

رایانه با سرعت بالا، حافظه‌ی بزرگ و قابلیت پردازش خودکار، ذخیره، جست‌وجو، ویرایش و حذف داده‌ها را بسیار آسان کرد. وجود رایانه سبب شد داده‌ها به شکل سازمان‌یافته (جدول، رکورد، فیلد) نگهداری شوند و پایگاه داده شکل بگیرد.

نگهداری اطلاعات به‌وسیله‌ی رایانه

اطلاعات در حافظه‌های جانبی (دیسک سخت، SSD، حافظه‌ی فلش) و امروزه در سرورها و فضای ابری ذخیره می‌شوند. مزیت‌ها: دسترسی سریع، حجم ذخیره‌سازی زیاد، کاهش مصرف کاغذ، امکان پشتیبان‌گیری و اشتراک‌گذاری در شبکه.

انفجار اطلاعات

یعنی رشد بسیار سریع حجم داده‌ها در دنیا (شبکه‌های اجتماعی، حسگرها، تراکنش‌های بانکی و...). این رشد باعث شده بدون پایگاه داده و ابزارهایی مانند داده‌کاوی، یافتن اطلاعات موردنیاز ممکن نباشد.

ابزارهای ذخیره‌سازی بشر

دیوار غارها و نقاشی، سنگ‌نوشته و لوح گلی، پاپیروس و پوست، کاغذ و کتاب، دفتر و پرونده‌های کاغذی، نوار و کارت‌های منگنه‌ای، دیسک‌های مغناطیسی و نوری (فلاپی، CD، DVD)، حافظه‌ی فلش، دیسک سخت و فضای ابری.

۲ صفحه ۴ کنجکاوی

چه فعالیت‌های دیگری را می‌توان نام برد که در آنها تعامل انسان با پایگاه داده وجود دارد. با دوستان خود گفت‌وگو کنید و حداقل دو مورد نام ببرید.

- خرید اینترنتی: ثبت و نمایش کالا، موجودی، سفارش و حساب کاربر در پایگاه داده‌ی فروشگاه.
- ثبت‌نام و دیدن کارنامه در سامانه‌ی مدرسه یا دانشگاه.
- رزرو بلیت قطار یا هواپیما و بررسی صندلی‌های خالی.

- جست‌وجو در موتورهای جست‌وجو، دیدن مخاطبین و پیام‌ها در تلفن همراه، شبکه‌های اجتماعی، نوبت‌دهی بیمارستان و دریافت قبض‌ها.

۳ صفحه ۴ فعالیت گروهی

مشکلی که در این روش جمع‌آوری اطلاعات وجود دارد چیست؟

هر فرد (معاون اجرایی، معاون آموزشی، هنرآموزان) اطلاعات یکسان هنرجویان را جداگانه از پرونده‌ها استخراج و ثبت می‌کند. مشکل این روش **تکرار داده‌ها (Redundancy)** است.

- وقت و نیروی انسانی بی‌جهت هدر می‌رود، زیرا یک کار چندبار انجام می‌شود.
- کاغذ و فضای ذخیره‌سازی بیشتری مصرف می‌شود.
- احتمال خطا و ناهمسانی اطلاعات بالا می‌رود.
- راه‌حل: اطلاعات یک بار ثبت شود و همه از یک منبع مشترک (پایگاه داده) استفاده کنند.

۴ صفحه ۵ فعالیت گروهی

مشکلات به‌وجود آمده در خصوص جمع‌آوری اطلاعات هنرجویان ناشی از چیست؟ چه راهکاری پیشنهاد می‌کنید؟ در این باره با هم‌کلاسی‌های خود گفت‌وگو کنید.

این مشکلات ناشی از آن است که هر فرد اطلاعات را به‌صورت جداگانه و در دفتر یا فهرست خودش نگهداری می‌کند؛ بنابراین سه مشکل پدید می‌آید:

- **تکرار داده‌ها:** یک اطلاعات چندبار ثبت می‌شود.
- **تداخل و مغایرت:** نمی‌توان تضمین کرد فهرست‌ها یکسان باشند (مثلاً نام‌خانوادگی کامل در یک فهرست و ناقص در فهرست دیگر).
- **دشواری بهروزرسانی:** برای اصلاح یک نام باید همه‌ی فهرست‌ها را اصلاح کرد و ممکن است کسی فراموش کند.

راهکار: ایجاد یک پایگاه داده‌ی مرکزی برای هنرستان. همه‌ی افراد از یک منبع مشترک داده می‌گیرند، هر اطلاعات فقط یک بار ثبت می‌شود و با یک بار اصلاح، در همه‌جا درست می‌شود. مسئول کتابخانه هم همیشه به اطلاعات به‌روز دسترسی دارد.

این فایل توسط سایت ایران‌توشه تهیه شده است.
irantooshe.ir

۵ صفحه ۶ کنجکاوی

دفترچه تلفنی را در نظر بگیرید که در آن نام و نام خانوادگی افراد و تلفن آنها پشت سر هم و بدون هیچ قاعده‌ای ثبت شده است. به نظر شما این روش چه مشکلی در پی خواهد داشت؟ دلیل خود را ذکر کرده، راه‌حل مورد نظر را ارائه دهید.

چون داده‌ها بدون ساختار و ترتیب نوشته شده‌اند:

- پیدا کردن یک شماره کند است، زیرا باید همه‌ی دفترچه را از ابتدا بخوانیم.
- نام، نام‌خانوادگی و تلفن از هم جدا نیستند؛ پس جست‌وجو و مرتب‌سازی بر اساس هرکدام ممکن نیست.
- ممکن است یک شخص چندبار ثبت شود و ویرایش یا حذف او دشوار است.

راه‌حل: داده‌ها را در جدولی با ستون‌های مشخص (نام، نام‌خانوادگی، تلفن) و به‌صورت مرتب‌شده بر اساس نام‌خانوادگی نگهداری کنیم؛ بهتر است برای هر فرد یک کد یکتا در نظر بگیریم و دفترچه را در یک پایگاه داده (مثلاً Access) ذخیره کنیم.

۶ صفحه ۷ فعالیت گروهی

محیط‌های عملیاتی پیرامون خود را شناسایی کنید و درباره‌ی آن با هم‌کلاسی‌های خود گفت‌وگو کنید.

محیط عملیاتی	مشکل و ضرورت استفاده از پایگاه داده
هنرستان	ثبت جداگانه‌ی اطلاعات هنرجویان، نمرات و حضور و غیاب؛ نیاز به گزارش سریع
کتابخانه	پیگیری امانت و بازگشت کتاب‌ها، جلوگیری از گم شدن اطلاعات اعضا
فروشگاه	کنترل موجودی کالا، صدور فاکتور و گزارش فروش
بانک	ثبت میلیون‌ها تراکنش، امنیت و دسترسی هم‌زمان چندین کاربر
داروخانه / بیمارستان	نگهداری پرونده‌ی بیماران و موجودی دارو، دقت بالا و دسترسی سریع
تاکسی تلفنی	ثبت سفارش مشتری‌ها و مدیریت رانندگان

۷ صفحه ۷ کنجکاوی

ـ آیا در محیطی مانند هنرستان، می‌توان میز و نیمکت‌ها را نیز به‌عنوان موجودیت در نظر گرفت؟

ـ برای پاسخ خود دلیل بیاورید.

ایران توشه
توشه‌ای برای موفقیت

ـ معیار انتخاب موجودیت چیست؟

بستگی به نیاز سیستم دارد. در پایگاه دادهای که کارش مدیریت هنرجویان، هنرآموزان و دروس است، میز و نیمکت موجودیت نیستند، زیرا نمی‌خواهیم درباره‌ی آنها اطلاعاتی ذخیره و بازیابی کنیم. اما اگر هدف سیستم «مدیریت اموال هنرستان» باشد، میز و نیمکت موجودیت هستند (با صفت‌هایی مانند کد اموال، تعداد، وضعیت و کلاس محل استقرار).

معیار انتخاب موجودیت: هر شخص، مکان، شیء یا مفهومی که در سناریوی محیط عملیاتی نقش دارد، قابل شناسایی از بقیه است و می‌خواهیم اطلاعاتی درباره‌ی آن ذخیره کنیم.

۸ صفحه ۸ پژوهش

در مورد تعریف موجودیت در منابع مختلف تحقیق کنید.

- **Entity** (موجودیت): چیزی در دنیای واقعی که وجود مستقل دارد، از دیگر اشیا قابل تشخیص است و می‌خواهیم داده‌هایی درباره‌ی آن نگهداری کنیم (مدل Chen).
- در مدل رابطه‌ای هر موجودیت به یک جدول تبدیل می‌شود؛ صفت‌های آن ستون‌ها و هر نمونه‌ی آن یک سطر (رکورد) است.
- موجودیت می‌تواند فیزیکی (هنرجو، کتاب، خودرو) یا مفهومی (درس، حساب بانکی، سفارش) باشد.
- موجودیت قوی بدون وابستگی به موجودیت دیگر وجود دارد؛ موجودیت ضعیف برای وجود خود به موجودیت دیگری وابسته است (مانند عضو خانواده‌ی کارمند).

۹ صفحه ۸ فعالیت کارگاهی

ـ با در نظر گرفتن محیط عملیاتی تاکسی‌تلفنی، جدول زیر را تکمیل کنید.

محیط عملیاتی | موجودیت

ایران توشه
توشه‌ای برای موفقیت

تاکسی‌تلفنی | راننده

محیط عملیاتی	موجودیت
تاکسی‌تلفنی	راننده
تاکسی‌تلفنی	مشتری (مسافر)
تاکسی‌تلفنی	خودرو (تاکسی)
تاکسی‌تلفنی	اپراتور
تاکسی‌تلفنی	سفر (درخواست)

۱۰ صفحه ۸ فعالیت کارگاهی

ـ موجودیت‌های محیط عملیاتی کتابخانه را نام ببرید؟

- عضو، کتاب، کتابدار، ناشر، نویسنده، موضوع (دسته‌بندی کتاب).
- امانت نیز در عمل یک ارتباط (یا جدول رابطه) میان عضو و کتاب است.

۱۱ صفحه ۸ فعالیت کارگاهی

جدول زیر را تکمیل کنید.

موجودیت | محیط عملیاتی | صفت‌ها

هنرجو

تاکسی تلفنی

مشتری فروشگاه

موجودیت	محیط عملیاتی	صفت‌ها
هنرجو	هنرستان	شماره‌ی دانش‌آموزی، نام، نام‌خانوادگی، کد ملی، تاریخ تولد، پایه، رشته، تلفن
راننده	تاکسی تلفنی	کد راننده، نام، نام‌خانوادگی، شماره‌ی گواهی‌نامه، پلاک خودرو، تلفن
مشتری	فروشگاه	کد مشتری، نام، نام‌خانوادگی، تلفن، نشانی

۱۲ صفحه ۹ کنجکاوی

ـ آیا می‌توان تمامی صفت‌های مربوط به یک موجودیت را در محیط عملیاتی مانند هنرستان در نظر گرفت؟

ایران توشه
توشه ای برای موفقیت

ـ معیار انتخاب صفت برای یک موجودیت چیست؟

ـ برای پاسخ خود دلیل بیاورید.

خیر. هر موجودیت صفت‌های بی‌شماری دارد (قد، وزن، گروه خونی، رنگ چشم، غذای مورد علاقه و...) و ذخیره‌ی همه‌ی آنها ممکن و لازم نیست؛ حجم پایگاه داده و کار ثبت داده را بی‌دلیل زیاد می‌کند.

معیار: فقط صفت‌هایی را انتخاب می‌کنیم که برای نیازهای اطلاعاتی آن محیط عملیاتی ضروری‌اند. در هنرستان قد و وزن هنرجو لازم نیست، اما نام، نام‌خانوادگی، شماره‌ی دانش‌آموزی و پایه لازم است. همچنین چون نام «علی» تکراری است و برای تشخیص یکتای هنرجو کافی نیست، صفتی مانند شماره‌ی دانش‌آموزی (کلید) لازم است.

۱۳ صفحه ۹ فعالیت گروهی

در جدول زیر برای هر محیط عملیاتی، یک موجودیت را انتخاب کرده، صفات آن را بنویسید. سپس یکی از این صفات را به‌عنوان کلید اصلی انتخاب کنید.

محیط عملیاتی | موجودیت | صفات موجودیت | صفت کلیدی

اداره

کتابخانه

محیط عملیاتی	موجودیت	صفات موجودیت	صفت کلیدی
اداره	کارمند	شماره‌ی کارمندی، نام، نام‌خانوادگی، کد ملی، سمت، تاریخ استخدام، حقوق	شماره‌ی کارمندی
کتابخانه	کتاب	شابک، عنوان، نویسنده، ناشر، موضوع، قیمت، سال انتشار	شابک (یا کد کتاب)

۱۴ صفحه ۱۰ پژوهش

تحقیق کنید استفاده از پایگاه داده چه معایبی دارد؟

- هزینه‌ی راه‌اندازی: خرید نرم‌افزار و سخت‌افزار مناسب، و آموزش یا استخدام متخصص.
- پیچیدگی طراحی و نگهداری، به‌ویژه برای پایگاه‌های بزرگ.
- تمرکز داده‌ها: اگر پایگاه داده خراب شود یا از کار بیفتد، کار همه‌ی کاربران متوقف می‌شود؛ پس پشتیبان‌گیری لازم است.

ایران توشه
توشه ای برای موفقیت

- آسیب‌پذیری امنیتی: دسترسی غیرمجاز به یک نقطه، همه‌ی داده‌ها را در معرض خطر قرار می‌دهد.
- برای کاربردهای کوچک و ساده (مثل یک فهرست کوتاه) پرهزینه و غیرضروری است.

۱۵ صفحه ۱۱ **فعالیت کارگاهی**

با در نظر گرفتن موجودیت‌های کشور و پایتخت، در جداول زیر هر کشور را به پایتخت خود متصل کنید. نتیجه‌گیری خود را از انجام این فعالیت بنویسید.

پایتخت | نام کشور

آنکارا | ایران

تهران | چین

پکن | ژاپن

توکیو | ترکیه

اتصال کشورها به پایتخت

نام کشور	پایتخت
ایران	تهران
چین	پکن
ژاپن	توکیو
ترکیه	آنکارا

نتیجه: هر کشور فقط یک پایتخت دارد و هر پایتخت فقط متعلق به یک کشور است؛ بنابراین ارتباط بین این دو موجودیت **یک‌به‌یک (1:1)** است.

ایران توشه
توشه ای برای موفقیت

۱۶ صفحه ۱۲ **فعالیت گروهی**

در جدول‌های زیر نام دو هنرآموز خود را مشخص کرده، برای هرکدام در جدول دیگر عنوان‌های درسی را در نظر بگیرید. سپس اتصال صحیح را انجام دهید. نتیجه‌گیری خود را از انجام این فعالیت بنویسید.

عنوان‌های درسی | نام هنرآموز

نمونه‌ی اتصال (نام‌ها فرضی است)

هنرآموز	عنوان‌های درسی
هنرآموز اول (آقای احمدی)	پایگاه داده، برنامه‌سازی
هنرآموز دوم (خانم کریمی)	شبکه، طراحی وب، سیستم‌عامل

نتیجه: هر هنرآموز می‌تواند چند عنوان درسی را تدریس کند، اما هر عنوان درسی (در این فرض) فقط به یک هنرآموز تعلق دارد. پس ارتباط هنرآموز با درس **یک‌به‌چند (N:1)** است: هر نمونه از موجودیت اول با چند نمونه از موجودیت دوم ارتباط دارد، ولی هر نمونه از موجودیت دوم تنها با یک نمونه از اول.

۱۷ صفحه ۱۳ کنجکاوی

به نظر شما چگونه می‌توانیم رابطه بین موجودیت‌ها و عملکردی که بین آنها در یک محیط عملیاتی وجود دارد را به‌صورت گرافیکی نشان دهیم و یک مدل معنایی ایجاد کنیم که بتواند در طراحی پایگاه داده و برقراری ارتباط ما را یاری کند؟

با رسم **نمودار موجودیت–ارتباط (ERD)**: موجودیت‌ها با مستطیل، صفت‌ها با بیضی (صفت کلیدی با زیرخط)، ارتباط‌ها با لوزی و نوع ارتباط (1 یا N یا M) روی خط‌های اتصال نمایش داده می‌شود. این نمودار مدل معنایی محیط عملیاتی است و پایه‌ی طراحی جدول‌ها و ارتباط بین آنها قرار می‌گیرد.

۱۸ صفحه ۱۳ فعالیت گروهی

در محیط عملیاتی کتابخانه در هنرستان یک نمونه ارتباط چند به چند را در نظر بگیرید و آن را به‌صورت دو جدول مجزا نمایش دهید سپس ارتباط‌های بین آنها را مشخص کنید. نتیجه‌گیری خود را از انجام این فعالیت بنویسید.

ارتباط «امانت» بین موجودیت‌های عضو و کتاب، چندبه‌چند است.

جدول عضو و کتاب‌هایی که امانت گرفته است

عضو	کتاب‌های امانت‌گرفته
علی احمدی	برنامه‌سازی #C، Access
سارا کریمی	برنامه‌سازی #C، شبکه
رضا مرادی	Access

جدول کتاب و اعضایی که آن را امانت گرفته‌اند

ایران توشه
توشه‌ای برای موفقیت

کتاب	اعضا
برنامه‌سازی C#	علی احمدی، سارا کریمی
Access	علی احمدی، رضا مرادی
شبکه	سارا کریمی

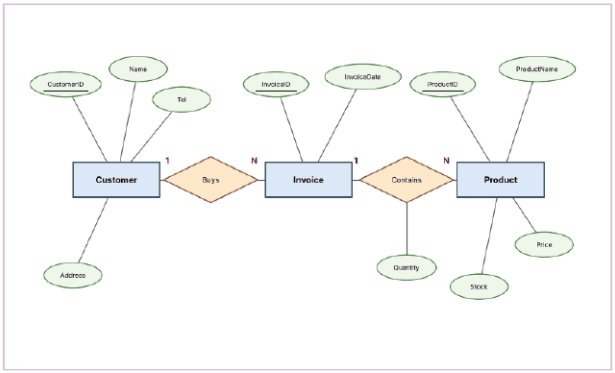
نتیجه: یک عضو می‌تواند چند کتاب را امانت بگیرد و یک کتاب نیز در زمان‌های مختلف توسط چند عضو امانت گرفته می‌شود؛ پس ارتباط **چندبه‌چند (N:M)** است. این ارتباط با دو جدول به‌تنهایی نمایش داده نمی‌شود و برای پیاده‌سازی به جدول سومی (جدول رابطه‌ی «امانت») نیاز داریم که کلید اصلی هر دو جدول را به‌عنوان کلید خارجی نگه می‌دارد.

۱۹ صفحه ۱۳ **فعالیت کارگاهی**

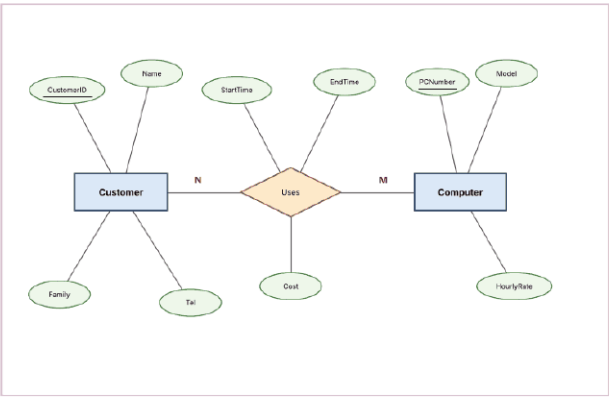
با در نظر گرفتن محیط‌های عملیاتی فروشگاه و کافی‌نت ارتباط موجودیت‌ها و نوع آنها را شناسایی کرده، نمودار آنها را رسم کنید.

ارتباط‌ها

محیط	موجودیت‌ها	ارتباط	نوع
فروشگاه	مشتری — فاکتور	خرید (Buys)	یک‌به‌چند (N:1)
فروشگاه	فاکتور — کالا	شامل (Contains)	چندبه‌چند (N:M)، با صفت تعداد
کافی‌نت	مشتری — رایانه	استفاده (Uses)	چندبه‌چند (N:M)، با صفت‌های ساعت شروع و پایان و هزینه



نمودار ER فروشگاه



نمودار ER کافی‌نت

ایران توشه
توشه ای برای موفقیت

۲۰ صفحه ۱۳ فعالیت گروهی

نام ارتباط یا عملکرد را در مثال‌های مطرح‌شده در ارتباطات چند به چند بالا شرح دهید؟ در این‌باره با هم‌کلاسی‌های خود گفت‌وگو کنید.

- هنرآموز و هنرستان: رابطه‌ی «تدریس در»؛ هنرآموز در چند هنرستان تدریس می‌کند.
- مترجم و کتاب: رابطه‌ی «ترجمه»؛ مترجم چند کتاب را ترجمه می‌کند و کتاب را چند مترجم ترجمه کرده‌اند.
- عضو و کتاب در کتابخانه: رابطه‌ی «امانت گرفتن».
- نام ارتباط معمولاً یک فعل است که عمل بین دو موجودیت را بیان می‌کند و داخل لوزی نوشته می‌شود.

۲۱ صفحه ۱۳ فعالیت گروهی

صفت شماره کارمندی و نام‌خانوادگی چه تفاوتی در نمودار دارند؟

شماره‌ی کارمندی **صفت کلیدی** است، زیرا برای هر کارمند منحصربه‌فرد است؛ در نمودار ER زیر آن **خط کشیده می‌شود**. نام‌خانوادگی ممکن است تکراری باشد و صفت معمولی است؛ پس بدون زیرخط نمایش داده می‌شود.

۲۲ صفحه ۱۴ فعالیت گروهی

موارد خواسته‌شده را به کمک هم‌کلاسی‌های خود انجام دهید.

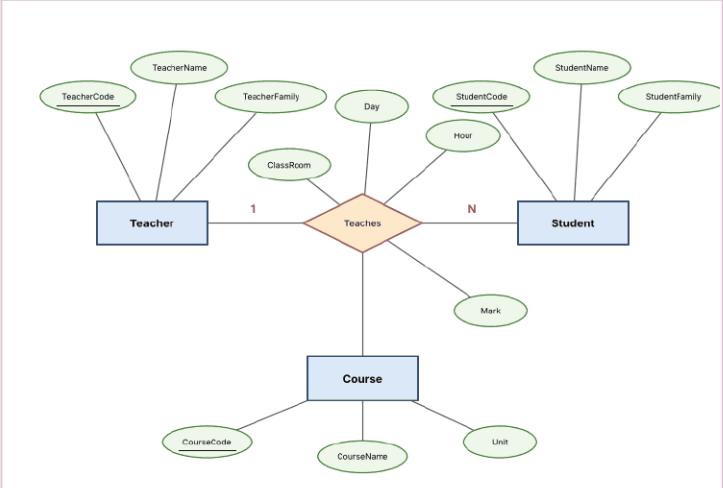
ـ در ارتباط تدریس بین هنرآموز و هنرجو، موجودیت درس و صفت‌های آن را تعریف کرده، با اضافه کردن آن، نمودار قبلی را کامل کنید.

ـ صفت نمره را در محل مناسب، به نمودار ER اضافه کنید.

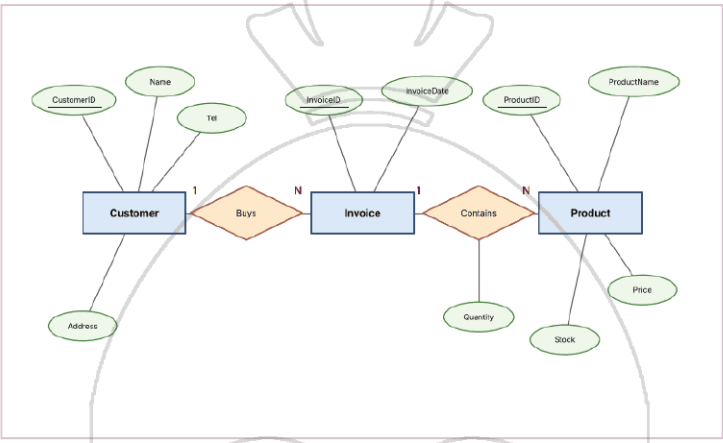
ـ برای محیط عملیاتی فروشگاه، نمودار ER رسم کنید.

- موجودیت درس (Course) با صفت‌های کد درس (کلید)، نام درس و تعداد واحد به نمودار اضافه و به لوزی «تدریس» وصل می‌شود.
- نمره مربوط به یک هنرجو در یک درس است، نه فقط به هنرجو یا درس؛ پس صفت **نمره** به **لوزی ارتباط** (کنار کلاس درس، روز درس و ساعت درس) وصل می‌شود.

ایران توشه
توشه‌ای برای موفقیت



نمودار ER فروشگاه (مشتری، فاکتور، کالا) در همین قسمت برای فعالیت قبل رسم شد:



۲۳ صفحه ۱۴ کنجکاوی

مثالی ارائه دهید که موجودیت با خودش در ارتباط باشد.

به این ارتباط، ارتباط بازگشتی (Recursive) می‌گویند. مثال‌ها:

- موجودیت کارمند: «مدیر بودن»؛ هر کارمند یک مدیر دارد و یک مدیر چند کارمند را مدیریت می‌کند (کارمند ← کارمند، N:1).
- موجودیت درس: «پیش‌نیاز بودن»؛ یک درس می‌تواند پیش‌نیاز چند درس دیگر باشد.
- موجودیت شخص: «ازدواج» (1:1).

ایران توشه
توشه‌ای برای موفقیت

۲۴ صفحه ۱۶ کنجکاوی

چند نمونه از سیستم‌های مدیریت پایگاه داده مشهور را نام ببرید.

MySQL، Oracle Database، Microsoft SQL Server، PostgreSQL، SQLite، MariaDB، IBM DB2، Microsoft Access و (از نوع غیررابطه‌ای) MongoDB.

۲۵ صفحه ۱۷ فعالیت گروهی

با شناسایی تعدادی از موجودیت‌های محیط عملیاتی هنرستان، فیلدهای آن را مشخص و در جدولی مانند شکل ۱۳ نمایش دهید.

جدول student

شماره دانش‌آموزی	نام	نام خانوادگی	پایه	رشته
856479665	حسین	عزیزی	یازدهم	شبکه و نرم‌افزار رایانه
968214763	علی	محمدی	دهم	الکترونیک

جدول teacher

کد کارمندی	نام	نام خانوادگی	رشته	تلفن
1011	علی	رنجبر	رایانه	34558896
1012	سارا	کریمی	الکترونیک	22334455

جدول course

کد درس	نام درس	تعداد واحد
5563	پایگاه داده	3
5564	برنامه‌سازی	4

هر ستون یک فیلد و هر سطر یک رکورد است.

۲۶ صفحه ۱۸ فعالیت کارگاهی

پایگاه داده‌ای با نام school در پوشه‌ای به نام DB ایجاد کنید.

1. نرم‌افزار Access را اجرا کنید و روی Blank desktop database کلیک کنید.

ایران توشه
توشه‌ای برای موفقیت

2. در کادر File Name نام school را بنویسید.
3. روی نماد پوشه‌ی کنار کادر کلیک کنید، در پنجره‌ی باز شده محل دلخواه (مثلاً درایو D) را انتخاب کنید و با دکمه‌ی New Folder پوشه‌ای به نام DB بسازید و آن را باز کنید.
4. OK و سپس Create را بزنید. پرونده‌ی school.accdb ساخته می‌شود و با یک جدول پیش‌فرض باز می‌شود.

۲۷ صفحه ۱۸ **پژوهش**

درباره‌ی تغییر قالب پرونده‌های Access از mdb به accdb تحقیق کنید و تفاوت‌های این دو قالب را بررسی کنید.

ویژگی	mdb (تا 2003)	accdb (از 2007)
موتور	Jet	ACE (جدیدتر)
رمزگذاری	ساده‌تر و ضعیف‌تر	قوی‌تر با رمز عبور پایگاه داده
نوع داده‌ی جدید	ندارد	Attachment، فیلد چندمقداری، فیلد محاسباتی و Long Text با قالب‌بندی
حجم و کارایی	بیشینه‌ی 2 گیگابایت	بیشینه‌ی 2 گیگابایت ولی کارایی و پایداری بیشتر
سازگاری	با نسخه‌های قدیمی هم باز می‌شود	فقط در Access 2007 به بعد باز می‌شود

۲۸ صفحه ۱۹ **کنجکاوی**

به نظر شما، دلیل وجود ستون Click to Add چیست؟

این ستون برای **افزودن سریع فیلد جدید** در نمای Datasheet است؛ با کلیک روی آن فهرستی از نوع داده‌ها باز می‌شود، نوع فیلد را انتخاب می‌کنیم و ستون جدیدی به جدول افزوده می‌شود که می‌توان نام آن را تایپ کرد. به این ترتیب بدون رفتن به نمای Design می‌توان ساختار جدول را گسترش داد.

۲۹ صفحه ۲۱ **فعالیت گروهی**

در جدول زیر با توجه به مقادیر هر فیلد، نوع داده‌ای مناسب را در ستون نوع فیلد انتخاب کنید.

ایران توشه
توشه‌ای برای موفقیت

نام فیلد | نوع فیلد | توضیحات | دلیل انتخاب نوع داده‌ای

نام فیلد	نوع فیلد	توضیحات	دلیل انتخاب
کد ملی	Short Text	شامل ارقام	عملیات ریاضی روی آن انجام نمی‌شود و صفر ابتدایی نباید حذف شود
نام خانوادگی	Short Text	نام خانوادگی کامل هنرجو	متن کوتاه (تا 255 نویسه)
رشته	Lookup Wizard	فهرست رشته‌ها با قابلیت انتخاب	مقدار از یک فهرست آماده انتخاب می‌شود
وضعیت قبولی	Yes/No	دو وضعیت قبول یا مردود	فقط دو مقدار منطقی دارد
نشانی	Long Text	نشانی محل سکونت	ممکن است طولانی‌تر از 255 نویسه باشد
تلفن	Short Text	تلفن ثابت هنرجو	صفر ابتدایی و خط تیره دارد و محاسبه نمی‌شود
تاریخ تولد	Date/Time	سال، ماه و روز	مقدار تاریخ است و امکان محاسبه‌ی سن دارد
عکس	Attachment	عکس پرسنلی هنرجو	پرونده‌ی تصویری ضمیمه می‌شود (OLE Object هم ممکن است)
رایانامه	Hyperlink	ذخیره‌ی نشانی رایانامه	پیوند mailto ساخته می‌شود (Short Text هم قابل قبول است)
معدل	Number	معدل به‌صورت اعشاری	عدد اعشاری (Field Size برابر Single یا Double)
حقوق	Currency	حقوق دریافتی کارمند	مقدار پولی با قالب پولی و دقت بالا

ایران توشه
توشه‌ای برای موفقیت

۳۰ صفحه ۲۲ **فعالیت کارگاهی**

یک پایگاه داده ایجاد کرده و جدول Register آن را با توجه به فیلدهای جدول زیر ایجاد کنید.

نام فیلد | نوع داده | ویژگی‌ها

MemberID | Number | اعداد صحیح، نمایش «کد عضویت» به جای نام فیلد

Name | Short Text | حداکثر 20 نویسه، نمایش «نام» به جای نام فیلد، ورود داده اجباری

Family | Short Text | حداکثر 25 نویسه، نمایش «نام خانوادگی» به جای نام فیلد، ورود داده اجباری

birth_of_date | Short Text | نمایش «تاریخ تولد» به جای نام فیلد، قالب ورود داده به‌صورت 0000/00/00

Average | Number | اعداد اعشاری، نمایش «معدل» به جای نام فیلد، عدد وارد شده در محدوده‌ی 0 تا 20 باشد در غیر این‌صورت پیغام خطای مناسبی نمایش داده شود.

Field | Short Text | نمایش «رشته تحصیلی» به جای نام فیلد، مقدار پیش‌فرض «رایانه» در این فیلد قرار گیرد.

در زبانه‌ی Create روی Table Design کلیک کنید و فیلدها را با تنظیمات زیر در Field Properties وارد کنید، سپس جدول را با نام Register ذخیره کنید.

نام فیلد	نوع داده	ویژگی‌های تنظیم‌شده
MemberID	Number	Field Size = Long Integer؛ Caption = کد عضویت (کلید اصلی)
Name	Short Text	Field Size = 20؛ Caption = نام؛ Required = Yes
Family	Short Text	Field Size = 25؛ Caption = نام خانوادگی؛ Required = Yes
birth_of_date	Short Text	Caption = تاریخ تولد؛ Input Mask = 0000/00/00
Average	Number	Field Size = Single؛ Decimal Places = 2؛ Caption = معدل؛ Validation Rule = >=0 And <=20؛ Validation Text = معدل باید عددی بین 0 تا 20 باشد.
Field	Short Text	Caption = رشته تحصیلی؛ Default Value = "رایانه"

این فایل توسط سایت ایران‌توشه تهیه شده است.
irantooshe.ir

۳۱ صفحه ۲۳ فعالیت کارگاهی

جدولی با نام student در نمای Design برای پایگاه داده school ایجاد کرده، فیلدهای مورد نیاز هنرجویان را در آن وارد کرده، نوع داده‌ای هر یک از این فیلدها را تعیین کنید.

ساختار جدول student

نام فیلد	نوع داده	توضیح
StudentCode	(Long Integer) Number	شماره‌ی دانش‌آموزی (کلید اصلی)
StudentName	(20) Short Text	نام
StudentFamily	(25) Short Text	نام خانوادگی
NationalCode	(10) Short Text	کد ملی
StudentTel	(12) Short Text	تلفن
BirthDate	Date/Time	تاریخ تولد
Grade	(10) Short Text	پایه

جدول را از زبانه‌ی Create ← Table Design می‌سازیم، نام و نوع هر فیلد را وارد می‌کنیم و با Ctrl+S با نام student ذخیره می‌کنیم.

۳۲ صفحه ۲۳ کنجکاوی

ـ چه روش دیگری برای تعیین کلید اصلی وجود دارد؟

ـ اگر جدول را بدون ایجاد کلید اصلی ذخیره کنید چه اتفاقی می‌افتد؟

- روش دیگر: راست‌کلیک روی ناحیه‌ی خاکستری سمت چپ نام فیلد در نمای Design و انتخاب Primary Key. در نمای Datasheet هم فیلد ID به‌طور خودکار کلید اصلی است.
- اگر بدون کلید اصلی ذخیره کنیم، Access پیامی می‌دهد: «There is no primary key defined...» و می‌پرسد آیا اکنون کلید اصلی ساخته شود. با Yes، فیلد ID از نوع AutoNumber به‌عنوان کلید اصلی افزوده می‌شود؛ با No جدول بدون کلید ذخیره می‌شود (ممکن است رکورد تکراری وارد شود و ارتباط‌دادن جدول مشکل باشد)؛ با Cancel ذخیره لغو می‌شود.

۳۳ صفحه ۲۴ کنجکاوی

پیام هشدار هنگام حذف فیلد شامل داده را به همراه ترجمه‌ی آن بنویسید.

```
Do you want to permanently delete the selected field(s) and all the data
in the field(s)? To permanently delete the field(s), click Yes.
```

ترجمه: «آیا می‌خواهید فیلد (های) انتخاب‌شده و همه‌ی داده‌های موجود در آن فیلد (ها) را برای همیشه حذف کنید؟ برای حذف دائمی فیلد (ها) روی Yes کلیک کنید.»

حذف فیلد قابل بازگشت (Undo) نیست، پس قبل از آن از پایگاه داده نسخه‌ی پشتیبان تهیه کنید.

۳۴ صفحه ۲۴ فعالیت کارگاهی

برای جدول student، کلید اصلی تعیین کنید.

1. جدول student را در نمای Design باز کنید.
2. روی فیلد StudentCode کلیک کنید (هر هنرجو شماره‌ی دانش‌آموزی یکتا دارد).
3. در زبانه‌ی Design، گروه Tools، روی Primary Key کلیک کنید تا نماد کلید کنار فیلد ظاهر شود و جدول را ذخیره کنید.

۳۵ صفحه ۲۵ فعالیت کارگاهی

در جدول student:

ـ فیلد جدید معدل برای نگهداری مقادیر اعشاری با دقت دو رقم اعشار را اضافه کنید.

ـ کلید اصلی جدول را به شماره ملی هنرجو تغییر دهید.

ـ فیلدهای غیرضروری را حذف و موارد دیگر را به جدول اضافه کنید تا فیلدهای دانش‌آموز تکمیل شود.

ـ رکوردهایی را به جدول اضافه کنید.

ـ نوع فیلد معدل را به Integer تغییر دهید و نتیجه‌ی این کار را در کلاس شرح دهید.

- **افزودن معدل:** در نمای Design فیلدی با نام Average و نوع Number اضافه کنید، Field Size را Single (یا Double) و Decimal Places را 2 بگذارید.
- **تغییر کلید اصلی:** فیلد NationalCode (از نوع Short Text، 10 نویسه) را اضافه کنید، روی آن راست‌کلیک و Primary Key را انتخاب کنید. کلید قبلی خودکار برداشته می‌شود.
- **حذف فیلدهای غیرضروری** (مثلاً قد و وزن) با Delete Rows و افزودن فیلدهای لازم (پایه، رشته، تاریخ تولد و...).
- **افزودن رکورد:** در نمای Datasheet داده‌ها را در سطر آخر وارد کنید.

نتیجه‌ی تغییر نوع معدل به Integer: فیلد فقط عدد صحیح نگه می‌دارد؛ بنابراین بخش اعشاری داده‌ها گرد می‌شود و از بین می‌رود (مثلاً 17.65 به 18 و 14.2 به 14 تبدیل می‌شود). Access قبل از ذخیره هشدار «Some data may be lost» می‌دهد. پس تغییر نوع فیلد حاوی داده، خطر از دست رفتن اطلاعات دارد و باید با احتیاط و پس از پشتیبان‌گیری انجام شود.

ایران توشه
توشه‌ای برای موفقیت

۳۶ صفحه ۲۵ فعالیت کارگاهی

جدول‌های زیر را در پایگاه داده school، ایجاد و جدول student را نیز طبق جدول مربوطه اصلاح کنید. زیر فیلد کلید اصلی در هر جدول خط کشیده شده است. بین جدول‌های teacher با student، teach با teach و course با teach، ارتباط N:1 برقرار است. در این محیط عملیاتی فرض شده است که هر درس به‌وسیله‌ی چندین هنرآموز تدریس می‌شود و برعکس (ارتباط N:M) و هر هنرجو چندین هنرآموز دارد و برعکس (ارتباط N:M).

جدول teacher: TeacherCode، TeacherName، TeacherFamily، TeacherTel

جدول course: CourseCode، CourseName، Unit

جدول student: StudentCode، StudentName، StudentFamily، StudentTel

جدول teach: RowID، CourseCode، TeacherCode، StudentCode، EducationYear، Mark

ساختار جدول‌ها (کلید اصلی با ★)

ایران توشه
توشه ای برای موفقیت

جدول	فیلد	نوع داده	توضیح
teacher	★ TeacherCode	Long) Number (Integer	کد کارمندی
	TeacherName	(20) Short Text	نام هنرآموز
	TeacherFamily	(25) Short Text	نام خانوادگی هنرآموز
	TeacherTel	(12) Short Text	تلفن هنرآموز
course	★ CourseCode	Long) Number (Integer	کد درس
	CourseName	(40) Short Text	نام درس
	Unit	(Byte) Number	تعداد واحد
student	★ StudentCode	Long) Number (Integer	شماره‌ی دانش‌آموزی
	/ StudentName StudentFamily	(25 / 20) Short Text	نام و نام خانوادگی
	StudentTel	(12) Short Text	تلفن
	StudentGrade	(10) Short Text	پایه (در کارگاه‌های بعد اضافه می‌شود)
teach	★ RowID	AutoNumber	ردیف
	CourseCode	Long) Number (Integer	کلید خارجی ← course
	TeacherCode	Long) Number (Integer	کلید خارجی ← teacher
	StudentCode	Long) Number (Integer	کلید خارجی ← student
	EducationYear	(9) Short Text	سال تحصیلی
	Mark	(Single) Number	نمره

جدول teach جدول رابطه است و رابطه‌ی N:M میان هنرآموز، هنرجو و درس را به سه ارتباط N:1 تبدیل می‌کند. معادل SQL (هر دستور را جداگانه در نمای SQL اجرا کنید؛ ابتدا سه جدول اصلی و سپس teach):

ایران توشه
توشه ای برای موفقیت

```
CREATE TABLE teacher (TeacherCode LONG PRIMARY KEY, TeacherName TEXT(20),
  TeacherFamily TEXT(25), TeacherTel TEXT(12))
CREATE TABLE course (CourseCode LONG PRIMARY KEY, CourseName TEXT(40), Unit BYTE)
CREATE TABLE student (StudentCode LONG PRIMARY KEY, StudentName TEXT(20),
  StudentFamily TEXT(25), StudentTel TEXT(12), StudentGrade TEXT(10))
CREATE TABLE teach (RowID COUNTER PRIMARY KEY, CourseCode LONG, TeacherCode LONG,
  StudentCode LONG, EducationYear TEXT(9), Mark SINGLE,
  CONSTRAINT fk1 FOREIGN KEY (CourseCode) REFERENCES course(CourseCode),
  CONSTRAINT fk2 FOREIGN KEY (TeacherCode) REFERENCES teacher(TeacherCode),
  CONSTRAINT fk3 FOREIGN KEY (StudentCode) REFERENCES student(StudentCode))
```

برای برقراری ارتباط: Database Tools ← Relationships، افزودن چهار جدول، کشیدن CourseCode، TeacherCode و StudentCode از جدول‌های اصلی روی فیلد هم‌نام در teach و فعال کردن Enforce Referential Integrity.

۳۷ صفحه ۲۸ فعالیت کارگاهی

پایگاه داده‌ای با نام Store ایجاد کنید و با توجه به موجودیت‌های موجود در فروشگاه، جدول‌های مورد نظر را برای آنها تعریف کرده و ارتباط بین جدول‌ها را برقرار کنید.

موجودیت‌ها: مشتری، کالا، فاکتور؛ ارتباط فاکتور و کالا چندبه‌چند است و با جدول InvoiceItem پیاده‌سازی می‌شود.

جدول‌ها (کلید اصلی با ★)

جدول	فیلدها
Customer	CustomerID (Number) ★، Address، Tel، Family، Name (همه Short Text)
Product	ProductID (Number) ★، ProductName (Short Text)، Price (Currency)، Stock (Number)
Invoice	InvoiceID (Number) ★، CustomerID (ک. خارجی)، InvoiceDate (Date/Time)
InvoiceItem	RowID (AutoNumber) ★، InvoiceID (ک. خارجی)، ProductID (ک. خارجی)، Quantity (Number)

```
CREATE TABLE Customer (CustomerID LONG PRIMARY KEY, Name TEXT(20), Family TEXT(25),
  Tel TEXT(12), Address TEXT(100))
CREATE TABLE Product (ProductID LONG PRIMARY KEY, ProductName TEXT(50),
  Price CURRENCY, Stock LONG)
CREATE TABLE Invoice (InvoiceID LONG PRIMARY KEY, CustomerID LONG, InvoiceDate DATETIME,
  CONSTRAINT fk_c FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID))
CREATE TABLE InvoiceItem (RowID COUNTER PRIMARY KEY, InvoiceID LONG, ProductID LONG,
  Quantity LONG,
  CONSTRAINT fk_i FOREIGN KEY (InvoiceID) REFERENCES Invoice(InvoiceID),
  CONSTRAINT fk_p FOREIGN KEY (ProductID) REFERENCES Product(ProductID))
```

سپس در پنجره‌ی Relationships هر کلید خارجی را روی کلید اصلی متناظر بکشید و Enforce Referential Integrity را فعال کنید؛ نوع داده‌ی کلید خارجی باید با کلید اصلی یکسان باشد.

۳۸ صفحه ۳۱ **کنجکاوی**

دو راه دیگر برای درج رکورد پیدا کنید.

- استفاده از **فرم** (Create ← Form) و واردکردن اطلاعات در آن.
- **درون‌ریزی داده** از پرونده‌ی دیگر مانند Excel یا متنی با External Data ← New Data Source یا Import و نیز کپی و چسباندن (Paste) رکوردها از Excel به نمای Datasheet.

۳۹ صفحه ۳۴ **فعالیت کارگاهی**

ـ به‌دلخواه یکی از جدول‌های موجود در پایگاه داده را بازکرده و سه رکورد جدید به روش مستقیم از طریق واسط کاربری برنامه در آن ایجاد کنید.

ـ با استفاده از دستور INSERT INTO، رکوردهای زیر را به جدول‌های teacher و student اضافه کنید.

جدول TeacherName، 1011 = teacher: TeacherCode = علی، TeacherFamily = رنجبر، TeacherTel = 34558896

جدول StudentName، 156 = student: StudentCode = حامد، StudentFamily = عزیزی، StudentTel = 35429902

روش مستقیم: جدول course را با دابل‌کلیک باز می‌کنیم و در سطر آخر (ستاره) سه رکورد وارد می‌کنیم؛ مثلاً (5563، پایگاه داده، 3)، (5564، برنامه‌سازی، 4) و (5565، شبکه، 2). با کلید Tab به فیلد بعدی می‌رویم.

با دستور SQL: زبانه‌ی Create ← Query Design ← SQL View و پس از نوشتن، دکمه‌ی Run. هر دستور را جداگانه اجرا کنید. (مقدارهای متنی داخل ' ' قرار می‌گیرند؛ نام‌های فارسی را هم می‌توان در همین قالب نوشت.)

```
INSERT INTO teacher (TeacherCode, TeacherName, TeacherFamily, TeacherTel)
VALUES (1011, 'Ali', 'Ranjbar', '34558896')

INSERT INTO student
VALUES (156, 'Hamed', 'Azizi', '35429902', '11')
```

ایران توشه
توشه‌ای برای موفقیت

۴۰ صفحه ۳۴ کارگاه ۲

ـ جدول teach را باز کرده و مقدار 5563 را برای فیلد کد درس، مقدار 1011 را برای فیلد کد کارمندی و مقدار 153 را برای فیلد شماره دانش‌آموزی وارد کنید. پس از ثبت این رکورد چه اتفاقی رخ می‌دهد؟

ـ امکان ورود مقادیر تکراری برای فیلدهای کلید خارجی را بررسی کنید.

ـ امکان خالی بودن فیلدهای کلید خارجی را بررسی نمایید.

- **ثبت رکورد:** چون Referential Integrity فعال است و در جدول‌های اصلی رکوردی با کد درس 5563 یا شماره‌ی دانش‌آموزی 153 نیست، Access رکورد را ذخیره نمی‌کند و پیام خطا می‌دهد: «You cannot add or change a record because a related record is required in table ...». رکورد زمانی ثبت می‌شود که ابتدا درس، هنرآموز و هنرجوی متناظر در جدول‌های اصلی وارد شده باشند.
- **مقدار تکراری:** مجاز است؛ چون فیلد کلید خارجی سمت «چند» ارتباط است (یک هنرآموز یا هنرجو می‌تواند در چندین رکورد teach بیاید). فقط مقدار کلید اصلی نباید تکراری باشد.
- **خالی بودن:** اگر Required = No باشد و فیلد جزو کلید اصلی نباشد، مقدار خالی (Null) پذیرفته می‌شود؛ ولی چنین رکوردی به هیچ رکورد اصلی وصل نیست. با Required = Yes خالی بودن ممنوع می‌شود.

۴۱ صفحه ۳۵ کنجکاوی

اگر از عبارت WHERE در دستور UPDATE صرف‌نظر شود، کدام رکوردهای جدول به‌روزرسانی خواهند شد؟

همه‌ی رکوردهای جدول. بدون WHERE شرطی برای انتخاب رکورد نیست، پس مقدار جدید در فیلد موردنظر همه‌ی رکوردها قرار می‌گیرد. مثلاً UPDATE student SET StudentGrade = '11' پایه‌ی همه‌ی هنرجویان را 11 می‌کند.

ایران توشه
توشه ای برای موفقیت

۴۲ صفحه ۳۵ فعالیت کارگاهی

با استفاده از دستور UPDATE:

ـ شماره تلفن هنرآموزی که در فعالیت‌های قبلی به جدول teacher اضافه شد را از 34558896 به 34557796 تغییر دهید.

ـ شماره دانش‌آموزی مربوط به هنرجویی با نام 'حامد عزیزی‌نیا' را به 53 تغییر دهید.

```
UPDATE teacher
SET TeacherTel = '34557796'
WHERE TeacherCode = 1011

UPDATE student
SET StudentCode = 53
WHERE StudentName = 'Hamed' AND StudentFamily = 'Azizinia'
```

در دستور دوم (نام فارسی را به‌جای 'Hamed' و 'Azizinia' بنویسید) اگر برای هنرجو رکوردی در جدول teach ثبت شده باشد، تغییر کلید اصلی فقط وقتی ممکن است که هنگام ایجاد ارتباط Cascade Update Related Fields را فعال کرده باشیم.

۴۳ صفحه ۳۶ کنجکاوی

چه روش دیگری برای حذف رکورد وجود دارد؟

- در نمای Datasheet روی نشانگر سطر (کادر خاکستری سمت چپ رکورد) راست‌کلیک کرده، Delete Record را انتخاب کنیم.
- رکورد را انتخاب کرده و در زبانه‌ی Home دکمه‌ی Delete را بزنیم.
- ساخت **Delete Query** در نمای Design پرس‌وجو (دستور DELETE) برای حذف چند رکورد با شرط.

۴۴ صفحه ۳۷ فعالیت کارگاهی

با استفاده از دستور DELETE :

ـ رکورد هنرآموزی با کد کارمندی 1011 را حذف کنید.

ـ اطلاعات تمامی هنرجویانی که شماره دانش‌آموزی آنها دو رقمی است را از جدول student حذف کنید.

```
DELETE
FROM teacher
WHERE TeacherCode = 1011

DELETE
FROM student
WHERE StudentCode >= 10 AND StudentCode <= 99
```

شرط دوم را با BETWEEN 10 AND 99 هم می‌توان نوشت. اگر برای این رکوردها در جدول teach رکوردی باشد، فقط هنگامی حذف می‌شوند که Cascade Delete Related Records فعال باشد، وگرنه Access مانع حذف می‌شود.

ایران توشه
توشه‌ای برای موفقیت

۴۵ صفحه ۳۷ **کارگاه ۳**

همه رکوردهای جدول را حذف کنید و نتیجه را بررسی کنید. آیا جدول حذف می‌شود؟

```
DELETE
FROM student
```

خیر. فقط داده‌ها (رکوردها) حذف می‌شوند و جدول خالی با همه‌ی فیلدها و ویژگی‌هایش باقی می‌ماند و می‌توان دوباره رکورد وارد کرد. برای حذف خود جدول باید از DROP TABLE student یا حذف از پنجره‌ی اشیا استفاده کرد. رکوردهای حذف‌شده قابل بازگشت نیستند؛ پس پیش از حذف نسخه‌ی پشتیبان تهیه کنید.

۴۶ صفحه ۴۰ **فعالیت کارگاهی**

با استفاده از wizard، یک پرس‌وجوی ساده ایجاد کنید به‌طوری‌که نام، نام خانوادگی و شماره تلفن هنرآموزان را نمایش دهد.

1. Create ← Query Wizard ← Simple Query Wizard ← OK.
2. از Tables/Queries جدول teacher را انتخاب کنید.
3. فیلدهای TeacherName، TeacherFamily و TeacherTel را با دکمه‌ی > به Selected Fields ببرید و Next را بزنید.
4. گزینه‌ی Detail را نگه دارید، نام پرس‌وجو را (مثلاً Teacher List) بنویسید و Finish کنید.

معادل SQL این پرس‌وجو:

```
SELECT TeacherName, TeacherFamily, TeacherTel
FROM teacher
```

ایران توشه
توشه ای برای موفقیت

۴۷ صفحه ۴۲ **فعالیت کارگاهی**

با استفاده از دستور SELECT یک پرس‌وجو ایجاد کنید که فهرستی از هنرجویانی که دروسی را اخذ کرده‌اند، نمایش دهد. نام و نام خانوادگی آنها در یک فیلد با هم ادغام‌شده و شماره دانش‌آموزی هنرجویان نیز در خروجی نمایش داده شود.

```
SELECT DISTINCT student.StudentCode, StudentName + ' ' + StudentFamily AS Student
FROM student, teach
WHERE teach.StudentCode = student.StudentCode
```

جدول teach فقط شماره‌ی هنرجویانی را دارد که درس گرفته‌اند؛ DISTINCT از تکرار هنرجو (به‌ازای هر درس) جلوگیری می‌کند. نمونه‌ی خروجی:

Student	StudentCode
Hamed Azizi	156
Reza Moradi	157
Mina Ahmadi	158

۴۸ صفحه ۴۳ فعالیت کارگاهی

مثال بالا را به‌گونه‌ای تغییر دهید که خروجی به‌صورت نزولی (DESC) مرتب شوند. سپس نتایج حاصل را با نتایج مثال قبل مقایسه کنید.

```
SELECT StudentCode, StudentName, StudentFamily
FROM student
ORDER BY StudentFamily DESC, StudentName DESC
```

مقایسه: در مثال قبل نام‌خانوادگی‌ها از A تا Z (صعودی، مثلاً Ahmadi، Azizi، Karimi، Moradi) و در این مثال برعکس، از Z به A (Moradi، Karimi، Azizi، Ahmadi) نمایش داده می‌شوند. اگر نام‌خانوادگی دو هنرجو یکسان باشد، ترتیب بر اساس نام تعیین می‌شود (در حالت نزولی نیز نام‌ها از آخر به اول).

۴۹ صفحه ۴۳ کارگاه ۶

جدول مورد استفاده در پرس‌وجو را باز کنید. آیا اطلاعات آن مرتب‌شده است؟

خیر. عبارت ORDER BY فقط ترتیب نمایش رکوردها در نتیجه‌ی همان پرس‌وجو را عوض می‌کند و ترتیب رکوردها در خود جدول (معمولاً بر اساس کلید اصلی) ثابت می‌ماند. برای مرتب کردن جدول باید از واسط کاربری Access (دکمه‌های Ascending و Descending) استفاده کرد.

۵۰ صفحه ۴۴ فعالیت کارگاهی

ـ با استفاده از واسط کاربری برنامه Access، جدول course را براساس کد درس به‌صورت نزولی مرتب کنید.

ـ با استفاده از دستورات SQL و عبارت ORDER BY، جدول course را براساس نام دروس و سپس تعداد واحد آنها به‌صورت صعودی مرتب کنید.

1. جدول course را در نمای Datasheet باز کنید.
2. روی عنوان ستون CourseCode کلیک کنید تا انتخاب شود.

3. در زبانه‌ی Home، گروه Sort & Filter، روی Descending کلیک کنید و جدول را ذخیره کنید.

با SQL:

```
SELECT *
FROM course
ORDER BY CourseName ASC, Unit ASC
```

۵۱ صفحه ۴۴ کارگاه ۷

با استفاده از دستورات SQL، یک پرس‌وجو شامل نام خانوادگی، تعداد واحد دروس اخذ شده و پایه تحصیلی هنرجو ایجاد کنید.

برای این منظور چه فیلدهایی را باید در مقابل دستور SELECT، قرار دهیم؟ SELECT ..... , ... , ...

این فیلدها متعلق به کدام جدول‌ها هستند؟ teach , ... , ... FROM

اگر بخواهیم تنها هنرجویانی نمایش داده شوند که برای آنها انتخاب واحد انجام شده است، به کدام جدول باید مراجعه کنیم؟

برای ایجاد شرط مناسب، عبارت زیر را کامل کنید. .... = student.StudentCode

```
SELECT StudentFamily, Unit, StudentGrade
FROM student, course, teach
```

- فیلدهای SELECT: StudentFamily، Unit و StudentGrade.
- StudentFamily و StudentGrade متعلق به جدول student و Unit متعلق به جدول course است؛ پس در FROM: student، course و teach.
- برای هنرجویانی که انتخاب واحد کرده‌اند باید به جدول teach مراجعه کنیم، چون رکورد هر درس‌گرفتن در آن ثبت می‌شود.
- شرط: teach.StudentCode = student.StudentCode

۵۲ صفحه ۴۵ کارگاه ۷

حال اگر بخواهیم تنها واحدهای دروسی که به‌وسیله‌ی هنرجویان اخذ شده‌اند را در نظر بگیریم، به کدام جدول باید مراجعه کنیم؟

آیا این جدول در مقابل FROM وجود دارد؟ برای ایجاد شرط مناسب، عبارت زیر را کامل کنید. ..... .

..... = course.CourseCode

ایران توشه
توشه‌ای برای موفقیت

اکنون دو شرط داریم که باید آنها را با یکدیگر ترکیب کرده و در مقابل دستور WHERE قرار دهیم. این دو شرط باید هم‌زمان بررسی شوند، در نتیجه با استفاده از عملگر AND آنها را با یکدیگر ترکیب کرده، دستور WHERE را تکمیل کنید. WHERE .....

به نظر شما چرا جدول teach را مقابل FROM قرار دادیم؟

- جدول course (چون Unit در آن است). بله، course در FROM آمده است.
- شرط دوم: teach.CourseCode = course.CourseCode

```
SELECT StudentFamily, Unit, StudentGrade
FROM student, course, teach
WHERE teach.StudentCode = student.StudentCode AND
      teach.CourseCode = course.CourseCode
GROUP BY StudentGrade, StudentFamily, Unit
```

چرا teach؟ جدول teach جدول رابطه است و کلید خارجی StudentCode و CourseCode را دارد؛ فقط از طریق آن می‌توان هنرجو را به درس‌هایش وصل کرد. بدون آن، ارتباطی میان student و course نیست و نتیجه‌ی پرس‌وجو بی‌معنا (ضرب دکارتی) می‌شود. هر فیلدی که در SELECT است و تابع تجمعی نیست، باید در GROUP BY هم بیاید.

۵۳ صفحه ۴۷ **فعالیت کارگاهی**

دستور SELECT را به‌گونه‌ای بنویسید که نام و نام‌خانوادگی هنرجویان را براساس پایه تحصیلی و نام درس گروه‌بندی کرده به‌صورت توأم نمایش دهد.

```
SELECT StudentGrade, CourseName, StudentName, StudentFamily
FROM teach, student, course
WHERE teach.StudentCode = student.StudentCode AND
      teach.CourseCode = course.CourseCode
GROUP BY StudentGrade, CourseName, StudentName, StudentFamily
```

نمونه‌ی خروجی

StudentFamily	StudentName	CourseName	StudentGrade
Ahmadi	Mina	Database	10
Ahmadi	Mina	Programming	10
Azizi	Hamed	Database	11
Moradi	Reza	Database	11

رکوردها ابتدا بر اساس پایه، سپس بر اساس نام درس و بعد نام هنرجو گروه‌بندی می‌شوند.

۵۴ صفحه ۵۰ کارگاه ۸

چگونه می‌توان فیلدهای سایر جدول‌ها یا پرس‌وجوها را به بخش فیلدهای انتخاب‌شده اضافه کرد؟

در همان مرحله‌ی انتخاب فیلدها، از فهرست کشویی Tables/Queries جدول یا پرس‌وجوی دیگر را انتخاب می‌کنیم؛ فیلدهای آن در Available Fields ظاهر می‌شود و با دکمه‌ی > یا >> به Selected Fields می‌رود. این کار را برای هر تعداد جدول می‌توان تکرار کرد (به شرط وجود ارتباط بین جدول‌ها). راه ساده‌تر این است که ابتدا پرس‌وجویی با همه‌ی فیلدهای لازم بسازیم و گزارش را روی آن بنویسیم.

۵۵ صفحه ۵۱ فعالیت کارگاهی

ـ با استفاده از wizard، گزارشی از فهرست هنرآموزان تهیه کنید.

ـ گزارشی از ثبت‌نام دانش‌آموزان به همراه تعداد واحدهای اخذ شده به‌وسیله‌ی هرکدام از آنها را که در مثال‌های قبل با استفاده از دستورات SQL انجام شد، تهیه کنید.

گزارش فهرست هنرآموزان: Create ← Report Wizard ← جدول teacher ← انتقال فیلدها با >> ← بدون گروه‌بندی ← مرتب‌سازی بر اساس TeacherFamily ← انتخاب Layout و جهت صفحه ← نام Teacher Report ← Finish.

گزارش واحدها: ابتدا پرس‌وجوی زیر را بسازید و با نام qryUnits ذخیره کنید:

```
SELECT StudentFamily, SUM(Unit) AS [Total Units], StudentGrade
FROM teach, student, course
WHERE teach.StudentCode = student.StudentCode AND
      teach.CourseCode = course.CourseCode
GROUP BY StudentGrade, StudentFamily
```

سپس Report Wizard را اجرا کنید، در Tables/Queries پرس‌وجوی qryUnits را انتخاب و همه‌ی فیلدها را منتقل کنید، گروه‌بندی را بر اساس StudentGrade بگذارید، مرتب‌سازی را بر اساس StudentFamily صعودی کنید و گزارش را با نام Units Report ذخیره کنید.

۵۶ صفحه ۵۳ فعالیت کارگاهی

گزارش ثبت‌نام دانش‌آموزان را که در فعالیت‌های قبل ایجاد کردید، در نمای Design بازکرده و موارد زیر را به آن اضافه کنید:

ـ شماره صفحه در سمت چپ پاصفحه قرارگرفته و تعداد کل صفحات را نیز نشان دهد.

ـ تاریخ و زمان را به گزارش اضافه کنید.

ـ لوگوی هنرستان را در گزارش قرار دهید.

ـ با استفاده از توابع تجمعی، تعداد رکوردها را در پاصفحه شمارش کنید.

ایران توشه
توشه‌ای برای موفقیت

ـ عنوان گزارش را با قلم و رنگ دلخواه در سر صفحه گزارش مشخص کنید.

روی برگه‌ی گزارش راست‌کلیک کنید و Design View را بزنید، سپس:

- **شماره‌ی صفحه:** Design ← Page Numbers؛ Format = Page N of M، Position = Bottom of Page (Footer)، Alignment = Left و OK. (معادل عبارت ="Page " & [Page] & " of " & [Pages] در یک Text Box پاصفحه.)
- **تاریخ و زمان:** Date and Time؛ گزینه‌های Include Date و Include Time را با قالب دلخواه فعال کنید.
- **لوگو:** Logo و انتخاب پرونده‌ی تصویر لوگوی هنرستان؛ در Report Header قرار می‌گیرد و با کشیدن جابه‌جا می‌شود.
- **شمارش رکوردها:** روی بخش Page Footer کلیک کنید، Totals ← Count Records (یا Text Box با مقدار =Count(*)).
- **عنوان:** روی برچسب عنوان در Report Header کلیک کنید و از زبانه‌ی Format نوع قلم، اندازه و رنگ (Font Color) را تعیین کنید.

۵۷ صفحه ۵۴ پروژه فاز ۱

می‌خواهیم برای کتابخانه‌ی یک هنرستان، پایگاه داده ایجاد کنیم. این پایگاه شامل موجودیت‌های عضو و کتاب است. با در نظر گرفتن این محیط عملیاتی موارد زیر را به ترتیب انجام دهید:

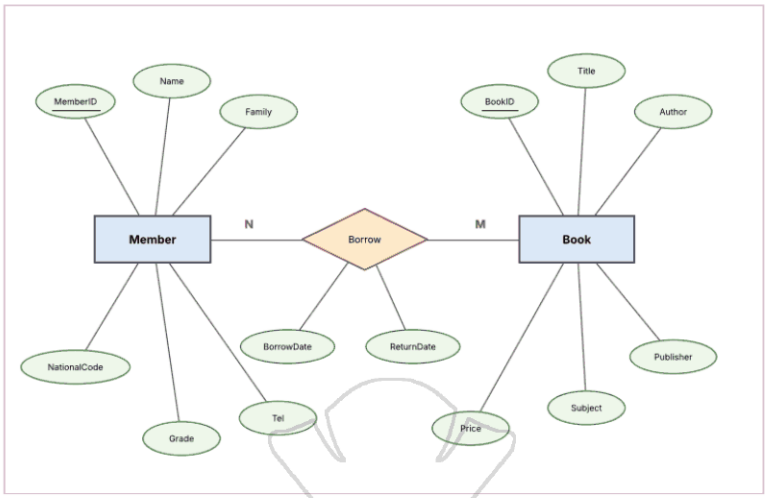
۱ـ موجودیت‌ها را شناسایی کرده، صفات و روابط بین آنها را بررسی کرده، نمودار ER آنها را به‌طور کامل رسم کنید.

موجودیت — صفات — کلید — ارتباط

موجودیت	صفات	کلید	ارتباط
عضو (Member)	MemberID، Name، Family، NationalCode، Grade، Tel	MemberID	امانت (Borrow) با کتاب: N:M
کتاب (Book)	BookID، Title، Author، Publisher، Subject، Price	BookID	امانت (Borrow) با عضو: N:M
امانت (Borrow)، صفت‌های ارتباط	BorrowDate (تاریخ امانت)، ReturnDate (تاریخ بازگشت؛ تا برگشت خالی)	BorrowID (در جدول)	کلید خارجی: MemberID و BookID

ایران توشه
توشه‌ای برای موفقیت

یک عضو می‌تواند چند کتاب را امانت بگیرد و یک کتاب در زمان‌های مختلف به چند عضو امانت داده می‌شود؛ پس ارتباط N:M است و در پیاده‌سازی به جدول سوم (امانت) تبدیل می‌شود. در نمودار، بیضی‌ها صفات (کلید با زیرخط)، مستطیل‌ها موجودیت‌ها و لوزی ارتباط را نشان می‌دهند.



۵۸ صفحه ۵۴ پروژه فاز ۱

۲ـ پایگاه داده را در Access ایجاد کنید.

۳ـ بر اساس نمودار ER، جدول‌های مورد نیاز را به همراه فیلدهای آنها ایجاد کنید و فیلد کلید اصلی و سایر جزئیات مربوط به فیلدها را در هر جدول تعیین کنید.

۴ـ ارتباط بین جدول‌ها را برقرار کنید.

ایجاد پایگاه داده: Blank desktop database با نام Library.accdb. سپس جدول‌ها را در نمای Design می‌سازیم:

ساختار جدول‌ها (کلید اصلی با ★)

ایران توشه
توشه ای برای موفقیت

جدول	فیلد	نوع و ویژگی
member	★ MemberID	Number (Long Integer)، Caption: کد عضویت
	Name / Family	Short Text (20 / 25)، Required = Yes
	NationalCode	Short Text (10)، Input Mask = 0000000000، Indexed بدون تکرار
	Grade / Tel	Short Text (10 / 12)
book	★ BookID	Number (Long Integer)
	Title	Short Text (80)، Required = Yes
	/ Author / Publisher Subject	Short Text (40 / 40 / 30)
	Price	Currency
borrow	★ BorrowID	AutoNumber
	MemberID	Number (Long Integer)، کلید خارجی ← member
	BookID	Number (Long Integer)، کلید خارجی ← book
	BorrowDate	Date/Time، Default Value = Date()
	ReturnDate	Date/Time، Required = No (تا بازگشت کتاب خالی)

```
CREATE TABLE member (MemberID LONG PRIMARY KEY, Name TEXT(20) NOT NULL,
  Family TEXT(25) NOT NULL, NationalCode TEXT(10), Grade TEXT(10), Tel TEXT(12))
CREATE TABLE book (BookID LONG PRIMARY KEY, Title TEXT(80) NOT NULL, Author TEXT(40),
  Publisher TEXT(40), Subject TEXT(30), Price CURRENCY)
CREATE TABLE borrow (BorrowID COUNTER PRIMARY KEY, MemberID LONG NOT NULL,
  BookID LONG NOT NULL, BorrowDate DATETIME, ReturnDate DATETIME,
  CONSTRAINT fk_m FOREIGN KEY (MemberID) REFERENCES member(MemberID),
  CONSTRAINT fk_b FOREIGN KEY (BookID) REFERENCES book(BookID))
```

ارتباط‌ها: Database Tools ← Relationships؛ جدول‌ها را اضافه کنید، member.MemberID را روی borrow.MemberID و book.BookID را روی borrow.BookID بکشید و Enforce Referential Integrity را فعال کنید (ارتباط 1:N از هر جدول اصلی به borrow)؛ این دو ارتباط با هم همان ارتباط N:M عضو و کتاب را می‌سازند.

۵۹ صفحه ۵۴ پروژه فاز ۱

۵ـ داده‌ها را در جدول‌ها وارد کنید.

ابتدا جدول‌های اصلی (member و book) و سپس borrow را پر می‌کنیم (وگرنه کنترل یکپارچگی اجازه‌ی ثبت نمی‌دهد). می‌توان در نمای Datasheet تایپ کرد یا از INSERT INTO استفاده کرد؛ بازگشت‌نداده‌ها با ReturnDate خالی ثبت می‌شوند:

```
INSERT INTO member VALUES (1001, 'Ali', 'Ahmadi', '0012345678', '11', '09121112233')
INSERT INTO member VALUES (1002, 'Sara', 'Karimi', '0023456789', '10', '09123334455')
INSERT INTO member VALUES (1003, 'Reza', 'Moradi', '0034567890', '12', '09125556677')
INSERT INTO member VALUES (1004, 'Mina', 'Rezaei', '0045678901', '11', '09127778899')

INSERT INTO book VALUES (1, 'C# Programming', 'Alavi', 'Fanni', 'Programming', 95000)
INSERT INTO book VALUES (2, 'Access 2016', 'Hosseini', 'Fanni', 'Database', 80000)
INSERT INTO book VALUES (3, 'Computer Networks', 'Tanenbaum', 'Ney', 'Network', 120000)
INSERT INTO book VALUES (5, 'Python Programming', 'Rahimi', 'Elm', 'Programming', 85000)

INSERT INTO borrow (MemberID, BookID, BorrowDate, ReturnDate)
VALUES (1001, 1, #2025-10-01#, #2025-10-15#)
INSERT INTO borrow (MemberID, BookID, BorrowDate)
VALUES (1001, 2, #2025-11-02#)
INSERT INTO borrow (MemberID, BookID, BorrowDate)
VALUES (1002, 5, #2025-11-05#)
INSERT INTO borrow (MemberID, BookID, BorrowDate, ReturnDate)
VALUES (1003, 3, #2025-10-20#, #2025-11-01#)
INSERT INTO borrow (MemberID, BookID, BorrowDate, ReturnDate)
VALUES (1002, 1, #2025-09-10#, #2025-09-20#)
```

۶۰ صفحه ۵۴ پروژه فاز ۱

۶ـ برای موارد خواسته شده پرس‌وجو ایجاد کرده، ذخیره کنید:

فهرست اعضا

فهرست کتاب‌ها

فهرست کتاب‌هایی با موضوع برنامه نویسی

فهرست اعضایی که کتابی را به امانت گرفته‌اند.

هر دستور را در نمای SQL یک Query Design جدید بنویسید، با Run اجرا کنید و با نام‌های qryMembers، qryBooks، qryProgBooks و qryBorrowers ذخیره کنید.

فهرست اعضا

ایران توشه
توشه ای برای موفقیت

```
SELECT MemberID, Name, Family, Grade, Tel
FROM member
ORDER BY Family, Name
```

فهرست کتاب‌ها

```
SELECT BookID, Title, Author, Publisher, Subject, Price
FROM book
ORDER BY Title
```

کتاب‌هایی با موضوع برنامه‌نویسی

```
SELECT BookID, Title, Author
FROM book
WHERE Subject = 'Programming'
```

اعضایی که کتاب امانت گرفته‌اند

```
SELECT DISTINCT member.MemberID, Name, Family
FROM member, borrow
WHERE member.MemberID = borrow.MemberID
```

نتیجه‌ی پرس‌وجوی آخر با داده‌های نمونه: (1001، Ali Ahmadi)، (1002، Sara Karimi)، (1003، Reza Moradi). عضو 1004 هرگز امانت نگرفته و در نتیجه نیست. DISTINCT از تکرار عضوی که چند کتاب گرفته جلوگیری می‌کند.

۶۱ صفحه ۵۴ پروژه فاز ۱

فهرست اعضایی که هنوز کتاب امانت گرفته شده را برنگردانده‌اند.

فهرست اعضا و تعداد کتاب‌هایی که تاکنون امانت گرفته‌اند.

فهرست کتاب‌های امانت گرفته شده به همراه اسامی کسانی که کتاب را امانت گرفته‌اند.

توشه‌ای برای موفقیت

اعضایی که کتاب را برنگردانده‌اند

```
SELECT DISTINCT member.MemberID, Name, Family
FROM member, borrow
WHERE member.MemberID = borrow.MemberID AND ReturnDate IS NULL
```

نمونه‌ی خروجی: Ali Ahmadi و Sara Karimi (کتاب‌هایشان هنوز بازنگشته است).

اعضا و تعداد کتاب‌های امانت‌گرفته

```
SELECT member.MemberID, Name, Family, COUNT(borrow.BookID) AS [Borrow Count]
FROM member, borrow
WHERE member.MemberID = borrow.MemberID
GROUP BY member.MemberID, Name, Family
```

نمونه‌ی خروجی

Borrow Count	Family	Name	MemberID
2	Ahmadi	Ali	1001
2	Karimi	Sara	1002
1	Moradi	Reza	1003

اگر بخواهیم اعضای بدون امانت نیز با عدد 0 بیایند، به‌جای FROM ... WHERE از FROM member LEFT JOIN borrow ON member.MemberID = borrow.MemberID استفاده می‌کنیم (عضو 1004 با تعداد 0 ظاهر می‌شود).

کتاب‌های امانت‌گرفته‌شده به همراه نام امانت‌گیرنده

```
SELECT Title, Name + ' ' + Family AS Member, BorrowDate
FROM book, borrow, member
WHERE borrow.BookID = book.BookID AND
      borrow.MemberID = member.MemberID
ORDER BY Title
```

نمونه‌ی خروجی

BorrowDate	Member	Title
2025-11-02	Ali Ahmadi	Access 2016
2025-10-01	Ali Ahmadi	C# Programming
2025-09-10	Sara Karimi	C# Programming
2025-10-20	Reza Moradi	Computer Networks
2025-11-05	Sara Karimi	Python Programming

این فایل توسط سایت ایران‌توشه تهیه شده است.
irantooshe.ir

۷ـ برای هر یک از پرس‌وجوهای مرحله‌ی قبل گزارشی تهیه و تنظیم کنید به گونه‌ای که در بالای هر گزارش نام کتابخانه به همراه لوگوی آن، تاریخ و ساعت گزارش قرار داشته باشد و شماره صفحه و مجموعه رکوردها در انتهای هر گزارش محاسبه شده باشد.

گزارش‌ها

گزارش	منبع	گروه‌بندی / مرتب‌سازی
Members Report	qryMembers	مرتب بر اساس Family
Books Report	qryBooks	گروه‌بندی بر اساس Subject
Programming Books	qryProgBooks	مرتب بر اساس Title
Borrowers Report	qryBorrowers	مرتب بر اساس Family
Not Returned Report	پرس‌وجوی ۵	مرتب بر اساس Family
Borrow Count Report	پرس‌وجوی ۶	مرتب بر اساس Borrow Count (نزولی)
Borrowed Books Report	پرس‌وجوی ۷	گروه‌بندی بر اساس Member

برای هر گزارش مراحل زیر را انجام می‌دهیم:

1. Create ← Report Wizard؛ در Tables/Queries پرس‌وجوی مربوط را انتخاب، فیلدها را منتقل، گروه‌بندی و مرتب‌سازی را تعیین، Layout را انتخاب و Finish کنید.
2. گزارش را در نمای Design باز کنید.
3. در Report Header: Logo (تصویر لوگوی کتابخانه)، Title (نام کتابخانه، مثلاً «کتابخانه‌ی هنرستان ...») و Date and Time را درج کنید.
4. در Page Footer: Page Numbers با قالب Page N of M را درج کنید.
5. رکوردها را بشمارید: روی Report Footer (یا Page Footer) کلیک کنید و Totals ← Count Records را بزنید (یا Text Box با مقدار =Count(*) و برچسب «تعداد رکوردها»).
6. با Print Preview نتیجه را ببینید و گزارش را ذخیره کنید.

برای اینکه نام و لوگو در همه‌ی گزارش‌ها یکسان باشد، یک گزارش را کامل کرده، با Save As کپی می‌کنیم و فقط Record Source آن را به پرس‌وجوی دیگر تغییر می‌دهیم.

ایران توشه
توشه‌ای برای موفقیت