

به نام خدا

توسعه برنامه و پایگاه داده

پاییز ۱۴۰۲

محمد جواد علی پور

فهرست

۳ واحد یادگیری ۱
۳ مفاهیم پایگاه داده
۶ RelationShip یا ارتباط
۱۱ کارگاه ۱: ایجاد پایگاه داده
۱۲ کارگاه ۲: صراحی جدول جدید
۱۴ کارگاه ۳: تغییر ساختار جدول
۱۵ Foreign key کلید خارجی
۱۵ ایجاد ارتباط بین جدول ها و ایجاد یکپارچگی
۱۶ کارگاه ۴: ایجاد ارتباط بین جدول ها
۱۷ واحد یادگیری ۲
۱۷ کارگاه ۱: اضافه کردن رکورد به جدول:
۱۷ دستور INSERT INTO
۱۸ کارگاه ۲: ویرایش رکوردها
۱۹ کارگاه ۳: حذف رکوردها:
۲۰ کارگاه ۴: ایجاد پرس و جو با استفاده از Wizard
۲۲ کارگاه ۶: مرتب سازی رکوردها
۲۲ کارگاه ۷: گروه بندی نتایج پرس و جو و استفاده از توابع جمعی:
۲۴ به این مثال ها توجه کنید:

موجودیت:

می شود. برای مثال، موجودیت کتاب می تواند دارای صفت های عنوان، شابک و قیمت باشد. موجودیت هنرآموز می تواند دارای صفت های شماره کارمندی، نام، نام خانوادگی، رشته، مدرک تحصیلی باشد.

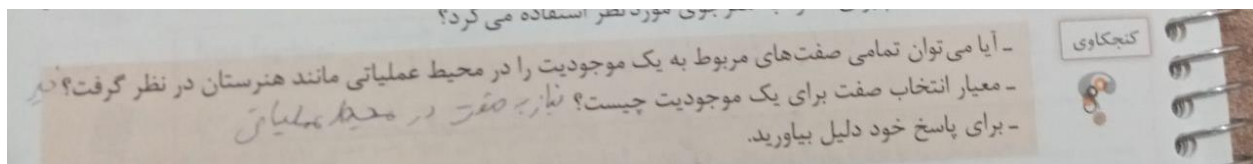
جدول زیر را تکمیل کنید.

موجودیت	محیط عملیاتی	صفت ها
هنرجو	هنرمندان	مسکن های فردی
		شماره تماس بهر و مادر
		کد پستی
		سال
راننده	تاکسی تلفنی	نام خانوادگی
		شماره تماس
		شماره پلاک
مشتری	فروشگاه	شماره تماس
		نام خانوادگی
		خدمات

- هر شخص و یا موجودیتی که و شیء (Object) بخواهیم درباره آن داده ذخیره کنیم موجودیت می گویم

صفت موجودیت:

- هر موجودیت دارای تعدادی صفت است، صفت برای توصیف ویژگی‌های یک موجودیت استفاده می‌شود



صفت کلیدی (Primary key)

- صفتی است که می‌توان با آن موجودیت را به صورت یکتا و منحصر به فرد شناسایی کرد.
 یا فقط مختص به صفت است و تکراری نیست

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

صفت کلیدی	صفات موجودیت	موجودیت	محیط عملیاتی
<input checked="" type="checkbox"/>	کد پرسنلی	پرسنل	اداره
<input type="checkbox"/>	نام و نام خانوادگی		
<input type="checkbox"/>	آدرس		
<input type="checkbox"/>	شماره تلفن		
<input type="checkbox"/>	شماره شناسنامه		
<input type="checkbox"/>	کتاب	کتابخانه	کتابخانه
<input type="checkbox"/>	دسته بندی		
<input type="checkbox"/>	مقدار		
<input type="checkbox"/>	نویسنده		
<input type="checkbox"/>	سال انتشار		

Relationship یا ارتباط

در محیط‌های عملیاتی ارتباطی منطقی بین موجودیت‌ها و صفت‌ها وجود دارد که به آنها Relational DataBase می‌گویند.

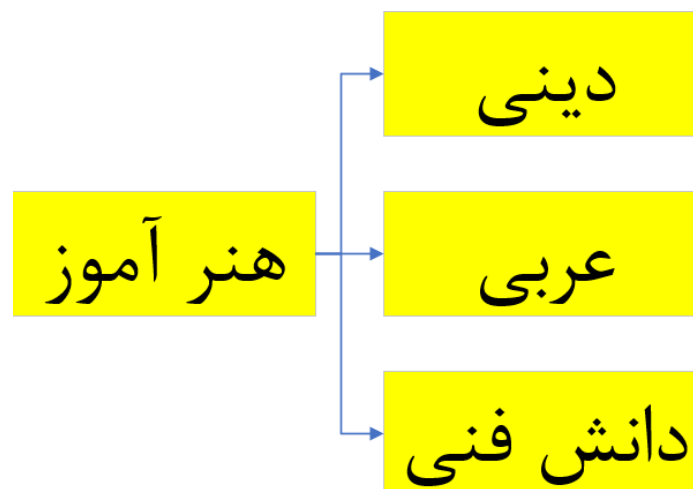
- ارتباط یک به یک (one to one)

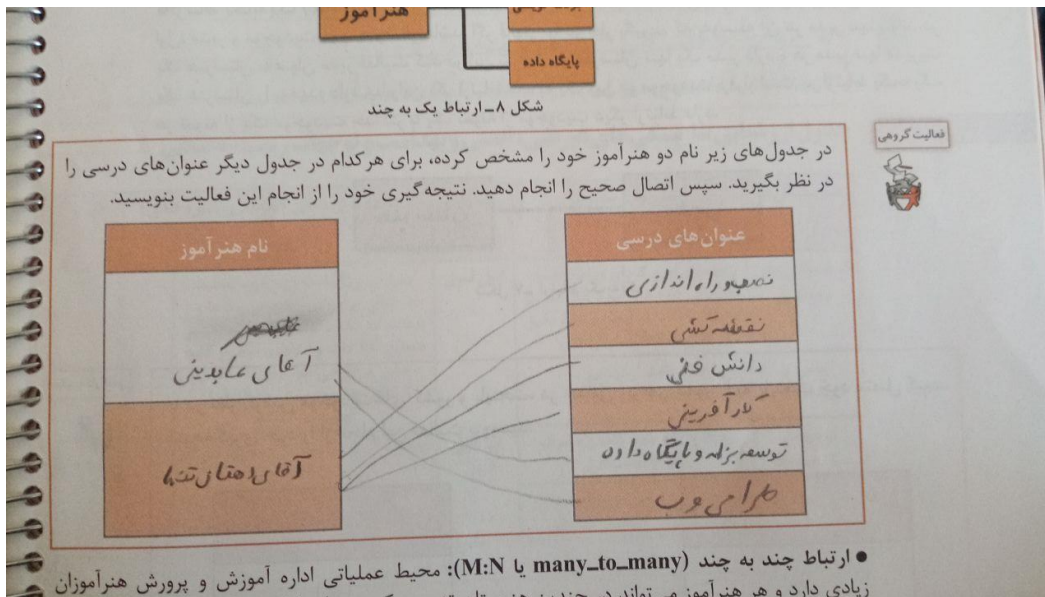
در ارتباط یک به یک، هر نمونه از یک موجودیت حداکثر با یک نمونه از موجودیت دیگر در ارتباط است



- یک به چند (one to many)

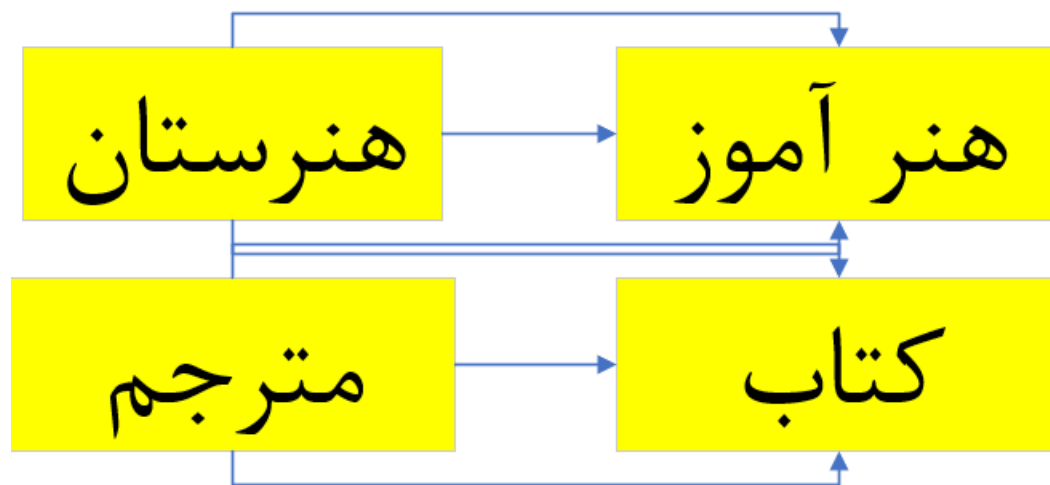
در این ارتباط هر یک نمونه از موجودیت اول ممکن است با چند نمونه از موجودیت دوم در ارتباط باشد، اما هر نمونه از موجودیت دوم فقط با یک نمونه از موجودیت اول در ارتباط است.





چند به چند (Many To Many)

در ارتباط چند به چند هر یک نمونه از موجودیت می تواند با چند نمونه از موجودیت دیگر در ارتباط باشد و بالعکس



نمودار رابطه بین موجودیت ها (ERD):

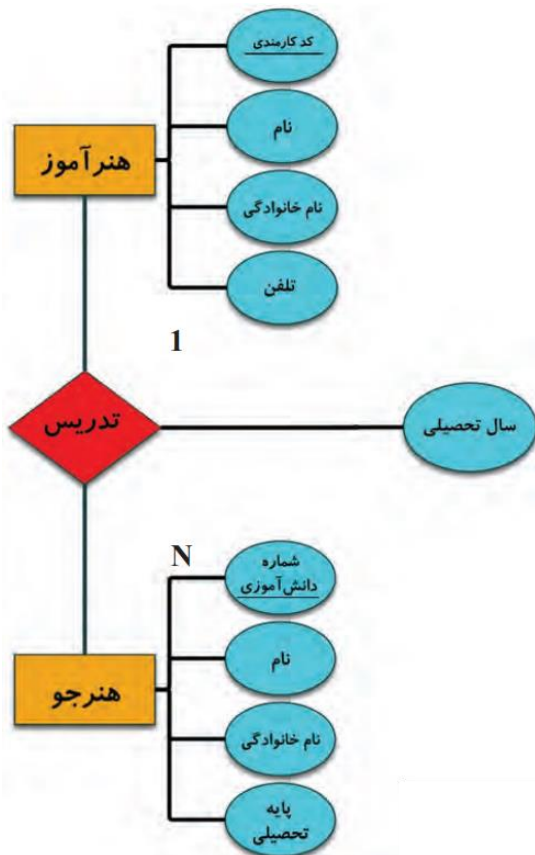
اولین قدم پس از شناسایی موجودیت ها و صفات آنها، تعیین ارتباط میان موجودیت ها است.

این کار با استفاده از ERD (Entity Relationship Diagram) به صورت گرافیکی نمایش داده می شود.

• **مستطیل:** برای نمایش موجودیت ها

• **بیضی:** برای نمایش صفات بین موجودیت ها

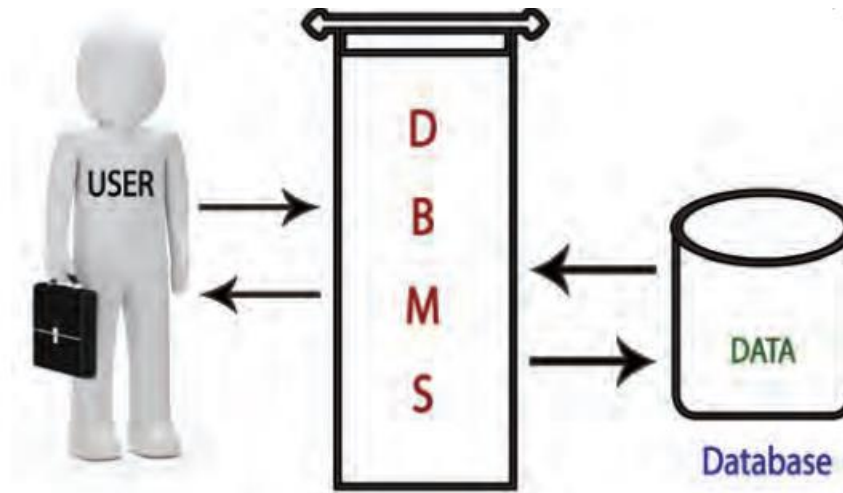
- **لوزی:** برای نشان دادن رابطه یا عملکرد بین موجودیت ها
- زیرخط: برای مشخص کردن صفت کلیدی



سیستم مدیریت پایگاه داده (DBSM)¹

پایگاه داده یکی از انواع روش های ذخیره و بازیابی اطلاعات است. (Information Storage and Retrieval)

سیستم مدیریت پایگاه داده واسط بین پایگاه داده و کاربران و نرم افزارها هستند



چند نمونه از سیستم های مدیریت پایگاه داده مشهور را نام ببرید.

کنجکاوی



MySQL, MariaDB, Microsoft SQL Server, Oracle DBMS

فیلد (Field) و رکورد (Record)

هر موجودیت دارای صفتهای مختلفی است که هر کدام از این صفتها فیلد نامیده می شود

محمدی = نام خانوادگی
مقدار فیلد نام فیلد

هر فیلد از دو قسمت تشکیل شده است:
نام فیلد و مقدار فیلد

یادداشت



هر فیلد دارای نوع داده‌ای خاصی است. به عبارت دیگر، نوع داده‌ای یک فیلد مشخص می‌کند چه نوع اطلاعاتی می‌تواند در آن قرار گیرد. مثال Text Short ، Text Long ، Number و... برای مثال در فیلد نام خانوادگی قرار است اطلاعات متنی قرار گیرد، در نتیجه نوع آن را Text Short در نظر خواهیم گرفت.

یادداشت



محمدی = نام خانوادگی
نام فیلد مقدار فیلد

هر فیلد از دو قسمت تشکیل شده است:
نام فیلد و مقدار فیلد

یادداشت



هر جدول با نام فیلدهایش تعریف می‌شود و نشان‌دهنده موجودیت یا ارتباط بین موجودیت‌ها است.

Access

Home

Home

New

Open

Blank database

ایجاد پایگاه داده جدید

جستجو برای نمونه های آنلاین

Search for online templates

Suggested searches: Database Business Logs Industry Lists Personal Contacts

We don't have any Office templates right now.

You can use the search box to find a specific template.

Not finding what you need?

Use the search box

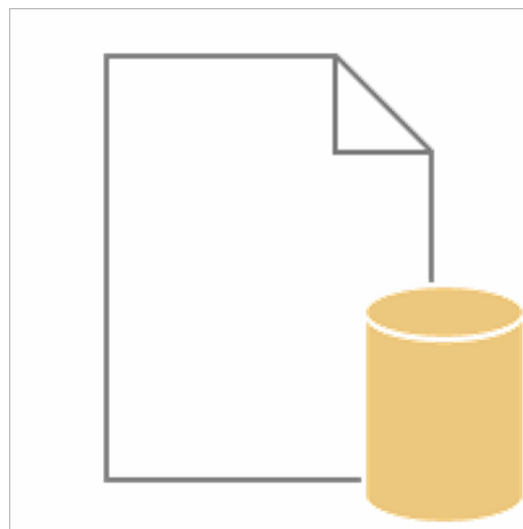
Account

Feedback

Options

باز کردن پایگاه داده موجود

زبان



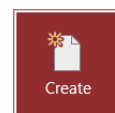
Blank database

File Name

Database2.accdb



C:\Users\javad\Documents\



قالب پیش فرض پرونده‌ها تا قبل از Access نسخه ۲۰۰۷، mdb بود، اما از نسخه ۲۰۰۷ به بعد به accdb تغییر یافت.

یادداشت



کارگاه ۲: صراحی جدول جدید

به اضافه کردن ستون جدید بدون رفتن به Desing view

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

کنجکاوی



در نمای Design جدول، امکان ایجاد جدول با جزئیات بیشتر وجود دارد.

یادداشت



بعد از تعیین نام داده باید نوع داده را مشخص کرد برای این کار در حالت Desing view در ستون Data type نوع داده را از لیست کشویی انتخاب می‌کنیم

جدول ۱- انواع داده‌ای فیلدها در Access2016

نوع داده	توضیحات
Short Text	این نوع داده ترکیبی از حروف و ارقام است که حداکثر تا ۲۵۵ نویسه در این نوع امکان پذیر است.
Long Text	این نوع داده، ترکیبی از حروف و ارقام است و حداکثر تا ۱GB نویسه را در خود ذخیره می کند.
Number	فقط اعداد را ذخیره می کند. ۱، ۲، ۴، ۸ و ۱۶ بایت
Date/Time	داده‌های مربوط به تاریخ میلادی و زمان را ذخیره می کند.
Currency	مقادیر پولی را ذخیره می کند.
AutoNumber	عددی که به صورت خودکار افزایش می یابد و با افزایش رکوردها، هر بار یک واحد به آن اضافه می شود.
Yes/No	این نوع داده می تواند مقادیر منطقی Yes/No یا True/False را ذخیره کند.
OLE Object	این نوع داده می تواند صدا، فیلم، عکس و نمودار را ذخیره کند.
Hyperlink	این نوع داده، پیوند به یک منبع اینترنتی را می پذیرد.
Attachment	این نوع داده به شما امکان می دهد که پرونده‌های خارجی را به پایگاه داده Access ضمیمه کنید.
Lookup Wizard	داده‌ها را از یک فهرست تایپ شده و یا سایر جداول نمایش می دهد.

فیلدها علاوه بر نام و نوع داده‌ای، دارای ویژگی‌های دیگری هستند که این ویژگی ها در بخش Filed properties قرار دارند

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

نام فیلد	نوع فیلد	توضیحات	دلیل انتخاب نوع داده‌ای
کد ملی هنرجو	Number	شامل ارقام	
نام خانوادگی	Short Text	نام خانوادگی کامل هنرجو	
رشته	Short Text	فهرست رشته‌های تحصیلی با قابلیت انتخاب رشته مورد نظر	
وضعیت قبولی	Yes/No	دو وضعیت قبول یا مردود	
نشانی	Long Text	نشانی محل سکونت هنرجو	
تلفن	Number	تلفن ثابت هنرجو	
تاریخ تولد	Date/Time	سال، ماه و روز	
عکس	OLE Object	عکس پرسنلی هنرجو	
رایانامه	Hyperlink	ذخیره نشانی رایانامه	
معدل	Number	معدل سال قبل هنرجو به صورت اعشاری	
حقوق	Currency	حقوق دریافتی یک کارمند	

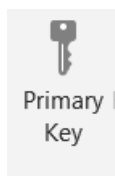
تنظیمات Filed properties برای نوع داده‌ای Short text

Field Properties	
General	Lookup
Field Size	255
Format	
Input Mask	
Action	
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	No
Indexed Compression	Yes
ReadOnly	No Control
ReadOnly Text	None
Text Align	General

The field description is optional. It helps you describe the field and is also displayed in the status bar when you select this field on a form. Press F1 for help on descriptions.

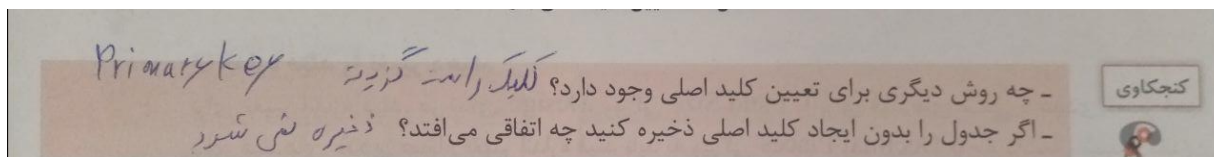
جدول ۲- ویژگی‌های رایج فیلدها

ویژگی	توضیحات
Field Size	تعداد نویسه‌های نوع فیلد Short Text را به تعداد مشخصی از ۱ تا ۲۵۵ نویسه محدود می‌کند. مقدار پیش‌فرض ۲۵۵ نویسه است. برای تعیین نوع دقیق داده عددی فیلد Number باید از ویژگی Field Size استفاده شود.
Format	قالب‌بندی نمایش داده در نمای Datasheet را تعیین می‌کند.
Input Mask	قالب داده در هنگام ورود اطلاعات را تعیین می‌کند. مثلاً برای ورود شماره تلفن ثابت قالب خاص ۰۰۰-۰۰۰۰۰۰۰۰ را می‌توان مشخص کرد. در این حالت کاربر ابتدا باید کد شهر و سپس شماره تلفن ثابت را وارد کند. نمونه داده: ۰۲۱-۱۲۳۴۵۶۷۸
Decimal Places	تعداد ارقام سمت راست نقطه اعشار را تعیین می‌کند. برای نوع داده‌ای Currency، Single، Double و Decimal که زیرمجموعه نوع Number هستند، استفاده می‌شود.
Caption	برچسب اختیاری است که به جای نام فیلد، در جدول، فرم‌ها و گزارش‌ها نمایش داده می‌شود.
Default Value	مقدار پیش‌فرضی است که در زمان ایجاد یک رکورد جدید، اگر مقداری به فیلد مربوطه اختصاص داده نشده باشد، در آن ذخیره می‌شود.
Validation Rule	شرطی است که برای ورود داده‌ها باید رعایت شود. به‌طور مثال، عدد وارد شده باید کمتر از ۱۰۰ باشد یا تاریخ وارد شده باید قبل از تاریخ روز جاری باشد.
Validation Text	متنی است که در صورت رعایت نکردن شرط Validation Rule به‌عنوان پیام خطا برای کاربر نمایش داده خواهد شد.
Required	مشخص می‌کند که در زمان ورود داده‌ها، حتماً باید برای این فیلد مقدار وارد شود و نمی‌تواند خالی رها شود.
Allow Zero Length	مشخص می‌کند که آیا می‌توان رشته خالی "" را برای فیلدی از نوع Short Text یا Long Text وارد کرد.
Index	باعث افزایش سرعت دسترسی اطلاعات خواهد شد. در صورتی که برای آن مقدار yes انتخاب شود، می‌توان تعیین کرد که مقدار تکراری در آن فیلد مجاز باشد یا خیر.



برای تعیین primary key دو راه وجود دارد

- یک بعد از انتخاب داده مدنظر وارد زبانه Design شد و از گروه Tools، primary key را انتخاب کرد
- بر روی داده کلیک راست کرده primary key را انتخاب کرده



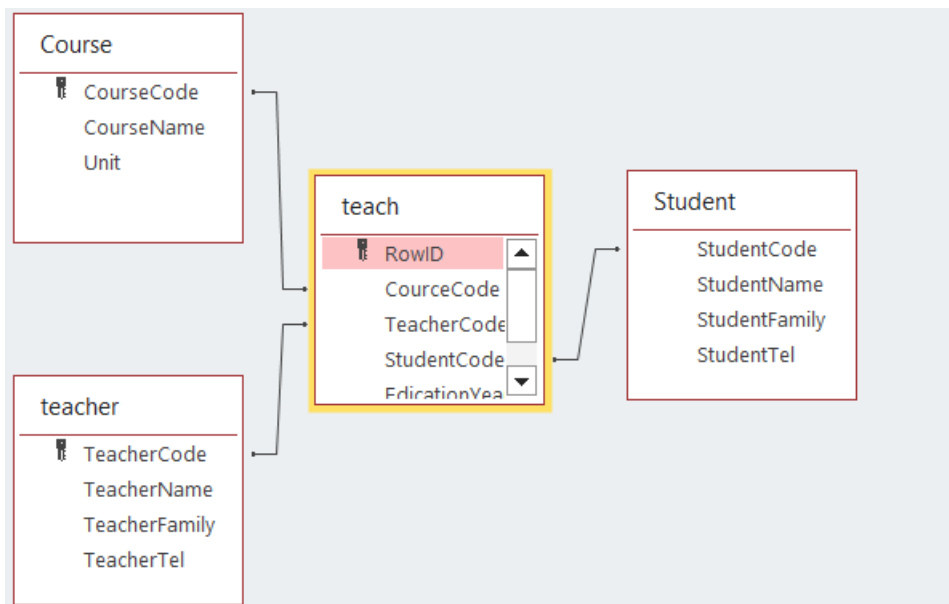
کارگاه ۳: تغییر ساختار جدول

برای اضافه کردن فیلدی به جدول در حالت Design view روی روی فیلد کلیک راست کرده و Insert Row را میزنم

برای حذف فیلد در حالت Design view روی فیلد کلیک راست کرده و گزینه Delete Rows را انتخاب می‌کنیم

کلید خارجی Foreign key

کلید خارجی به فیلدی از جدول گفته می‌شود که به صورت یکتا و منحصر به فرد در جدول دیگر مشخص می‌شود به عبارت دیگر، کلید خارجی در جدول دوم تعریف می‌شود ولی به کلید اصلی در جدول خودش وصل است.



ایجاد ارتباط بین جدول‌ها و ایجاد یکپارچگی

قبل از ایجاد ارتباط میان جدول‌ها باید بین آنها یکپارچگی وجود داشته باشد. برای مثال در جدولی اطلاعات هنرجو نگه داری می‌شود و در جدول دیگر جزئیات کلاس اگر یکی از این داده‌ها حذف شود بقیه داده‌های مربوط هم باید حذف شوند در غیر این صورت داده‌ها بالاتکلیف می‌مانند.

در Access این قابلیت با نام Referential Integrity نامیده می‌شود که باعث ایجاد این هماهنگی می‌شود.

برای اینکه این قابلیت به درستی اعمال شود باید شرایط زیر وجود داشته باشند:

- هر دو جدول متعلق به یک پایگاه داده باشند
- فیلهای برقرار کننده ارتباط در هر دو جدول، دارای نوع داده‌ای یکسان باشند
- فیلد برقرار کننده ارتباط در جدول اصلی، به عنوان کلید اصلی تعریف نشده باشد

اگر بخواهیم رکوردی را از جدول اصلی حذف کنیم باید تیک گزینه Cascade Delete Related Records را هنگام ایجاد ارتباط انتخاب کرده باشیم با این کار با حذف رکوردی تمام اطلاعات مربوط حذف می‌شود.

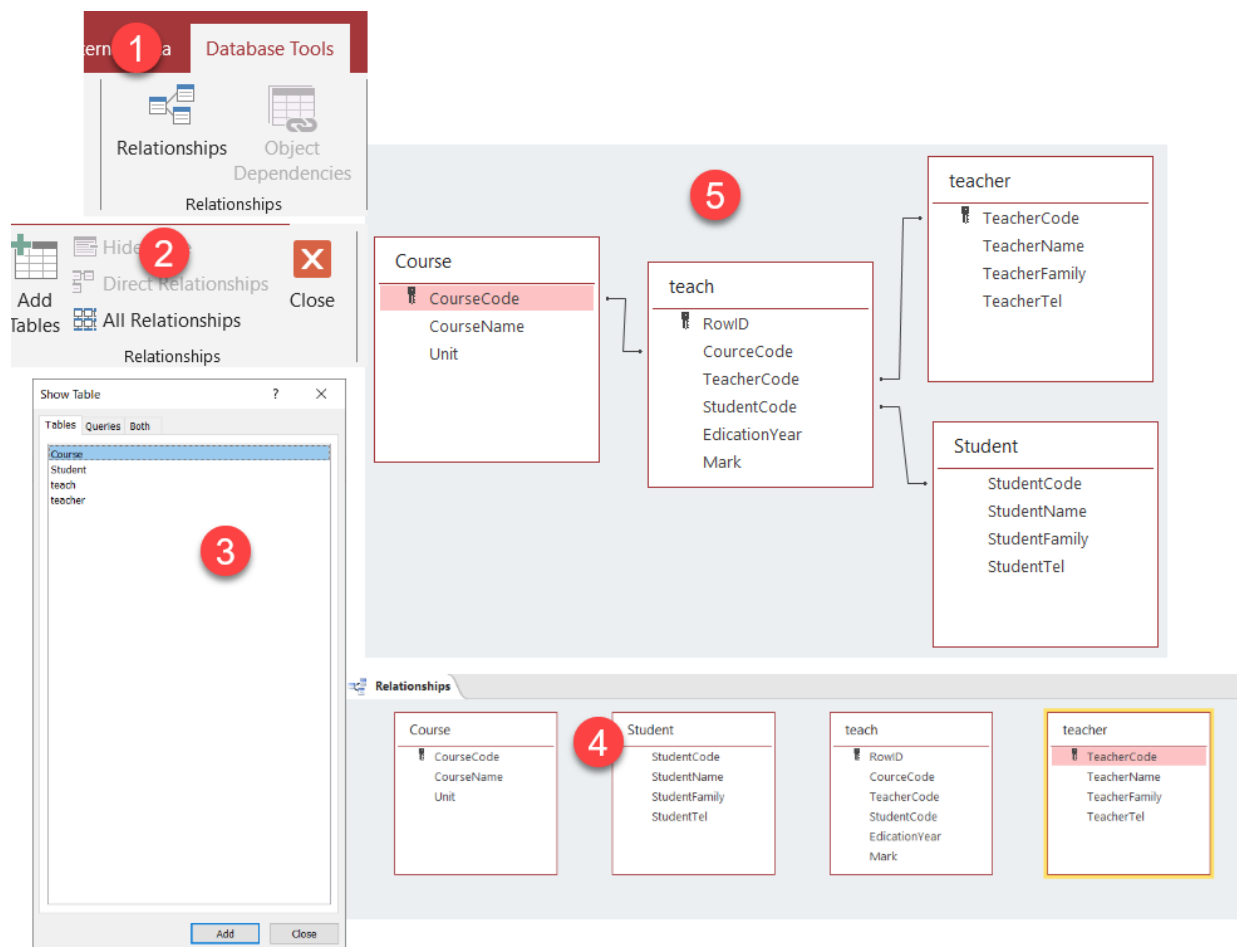
برای به روزرسانی مقادیر فیلد اصلی باید در هنگام ایجاد ارتباط گزینه Cascade Update Related Fields را بزنیم در این صورت تغییرات فیلد کلید اصلی بر روی تمامی فیلهای در جدول‌های دیگر اعمال می‌شود.

کارگاه ۴: ایجاد ارتباط بین جدول‌ها

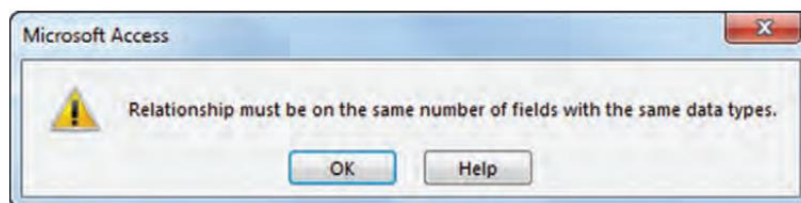
بعد از وارد کردن اطلاعات در جدول‌ها طبق قاعده گفته شده تمامی جدول‌ها را ذخیره کرده و می‌بندیم.

برای ایجاد ارتباط به زبانه DataSheet Tools رفته و گروه Relationships گزینه Relationships کلیک می‌کنیم. کادر Show Table نمایش داده می‌شود اگر نمایش داده نشد در زبانه Relationships Design گزینه Add Table را می‌زنیم و جدول‌ها را انتخاب و اضافه می‌کنیم.

برای اتصال جدول‌ها به یکدیگر با ماوس فیلد مدنظر را انتخاب کرده و وصل می‌کنیم به فیلد مرتبط در جدول دیگر.



فیلدی که به عنوان کلید خارجی در نظر گرفته می‌شود، باید دارای همان نوع داده‌ای باشد که برای کلید اصلی در جدول اصلی در نظر گرفته شده است. در صورتی که فیلدهای به هم مرتبط دارای نوع داده متفاوت باشند با پیام خطا مواجه خواهید شد (شکل ۲۴).



شکل ۲۴- پیام خطای عدم مطابقت نوع داده‌ای فیلدهای کلید خارجی و کلید اصلی

کارگاه ۱: اضافه کردن رکورد به جدول:

۱- با تایپ مقادیر فیلدها در ردیف آخر رکوردها (علامت *) یک رکورد جدید ایجاد می‌شود.
۲- کلیک روی دکمه NewRecord در پایین نمای Datasheet

دو راه دیگر برای درج رکورد پیدا کنید.

کنجکاوی

SQL² زبانی است که برای ارتباط با پایگاه داده استفاده می‌شود با استفاده از SQL می‌توان تمامی قابلیت‌های رابط کاربری مانند Delete و Update چند قابلیت بیشتر را اجرا کرد

• SQL به کوچک و یا بزرگ بودن حروف حساس نیست اما برای خوانایی بیشتر دستورات را با حروف بزرگ می‌نویسند.

برای ایجاد دستورات جدید در Access وارد زبانه Create می‌شویم از گروه Quarries گزینه Query Design را انتخاب می‌کنیم اگر Show Table باز شد آن را می‌بندیم و وارد SQL View می‌شویم.

دستور INSERT INTO

شکل کلی دستور INSERT INTO

```
INSERT INTO table_name (Filed First, Filed_Second, Field_third, .....)  
VALUES (value1, value2, value3, .....)
```

جدول ۳ - بخش‌های مختلف دستور INSERT INTO

بخش	توضیحات
table_name	نام جدولی که قرار است رکوردها به آن اضافه شوند.
field1,field2	نام فیلدهایی که قرار است، داده‌ها در آنها قرار گیرند.
value1,value2	مقادیری که درون فیلدهای متعلق به رکورد جدید قرار می‌گیرند. هر مقدار در فیلد متناظر خود که در ابتدای دستور مشخص شده است قرار می‌گیرد. برای مثال، value1 در field1 قرار می‌گیرد. مقادیر با استفاده از کاما (,) از یکدیگر جدا شده و فیلدهای متنی درون علامت نقل قول (') قرار می‌گیرند.

یادداشت



در شکل کلی دستورات SQL استفاده از عبارت‌های داخل [] اختیاری است.

● درج یک رکورد:

```
INSERT INTO teacher (TeacherCode, TeacherFamily, TeacherTel)
VALUES (145231,'احمدی','66085699')
```

● درج یک رکورد با مقادیر همه فیلدها

```
INSERT INTO teacher
VALUES (145231,'علی','محمدی','66085699')
```

یادداشت



در زمان نوشتن دستورات SQL، مقادیر رشته‌ای باید حتماً در داخل '' قرار بگیرند.

کارگاه ۲: ویرایش رکوردها

شکل کلی دستور UPDATE

UPDATE table_name

SET newvalue

WHERE criteria

جدول ۴- بخش‌های مختلف دستور UPDATE

بخش	توضیحات
table	نام جدولی که قصد تغییر اطلاعات آن را داریم.
newvalue	مقادیر جدیدی که قرار است در فیلدهای خاصی از رکورد به‌روزرسانی شده، قرار گیرد.
criteria	عبارتی که مشخص می‌کند چه رکوردهایی به‌روزرسانی شوند. تنها رکوردهایی که دارای این معیار هستند، به‌روزرسانی خواهند شد.

در مثال زیر، نام خانوادگی هنرجویی با شماره دانش‌آموزی ۱۵۶ را به 'عزیزی نیا' تغییر می‌دهیم:

```
UPDATE student
SET StudentFamily = 'عزیزی نیا'
WHERE StudentCode = 156
```

کنجکاوی



اگر از عبارت WHERE در دستور UPDATE صرف‌نظر شود، کدام رکوردهای جدول به‌روزرسانی خواهند شد؟ **تمام رکوردها به روز می‌شود**

کارگاه ۳: حذف رکوردها:

هنگامی که رکوردی حذف می‌شود، تمامی داده‌های موجود در فیلدهای آن رکورد نیز حذف خواهد شد. این فرایند سبب حذف کل آن ردیف از جدول می‌شود. رکوردها را می‌توان به صورت دستی و تک به تک حذف کرد و یا با استفاده از دستور DELETE، تعداد زیادی رکورد را به یک باره حذف کرد.

چه روش دیگری برای حذف رکورد وجود دارد؟ پاسخ: با کلیک راست روی سطر مورد نظر در نمای Datasheet و انتخاب گزینه Delete Record می‌توان یک رکورد را حذف نمود.

کنجکاوی



یادداشت



هنگامی که رکوردی حذف می‌شود، نمی‌توان آن را دوباره بازگرداند. به همین دلیل، همیشه قبل از حذف داده‌ها، از پایگاه داده خود نسخه پشتیبان تهیه کنید.

شکل کلی دستور DELETE

```
DELET FROM table_name  
WHERE criteria
```

جدول ۵- بخش‌های مختلف دستور DELETE

بخش	توضیحات
table_name	نام جدولی که قرار است رکوردهای آن را حذف کنید.
criteria	عبارتی که تعیین می‌کند چه رکوردهایی از جدول حذف شوند.

به مثال زیر توجه کنید:

- حذف رکورد هنرجویی با شماره دانش‌آموزی ۱۵۶ از جدول student:

```
DELETE FROM student
```

```
WHERE StudentCode = 156
```

- حذف کل رکوردهای یک جدول

```
DELETE FROM student
```

کارگاه ۴: ایجاد پرس و جو با استفاده از Wizard

برای ایجاد از زبانه Create گروه Queies گزینه Query Wizard کلیک کرده از کادر باز شده simple Query wizard را انتخاب می‌کنیم.

۱. از فهرست کشویی Table/Queries جدول مورد نظر را اضافه می‌کنیم.
۲. فیلدهایی که می‌خواهیم لحاظ شوند انتخاب می‌کنیم
۳. و با زدن روی دکمه های >> یا > فیلدها را انتخاب می‌کنیم
۴. در آخر نام Wizard را وارد کرده و بعد روی دکمه Finish کلیک می‌کنیم

کارگاه ۵: دستور SELECT

شکل کلی دستور SELECT

```
SELECT field_list FROM table_name
WHERE ...
GROUP BY ....
ORDER BY ....
```

جدول ۶- بخش‌های مختلف دستور SELECT

بخش	توضیحات
field – list	نام فیلدهای حاوی داده است که می‌خواهیم آنها را بازیابی کنیم. اگر بیش از یک فیلد مشخص شده باشد، با همان ترتیب قرار گرفته‌شده، بازیابی می‌شوند.
table – name	نام جدولی که حاوی فیلدهایی است که رکوردها از آن انتخاب می‌شوند.

شکل ساده دستور SELECT

```
SELECT fields FROM table
```

کارگاه ۵: ایجاد پرس‌وجو با استفاده از دستور SELECT

- تمامی فیلدهای جدول student انتخاب خواهد شد:

```
SELECT * FROM student
```

- مشاهده فهرستی از اسامی هنرجویان به همراه شماره دانش آموزی آنها:

```
SELECT StudentCode, StudentName, StudentFamily
FROM student
```

- مشاهده فهرست دروس به همراه تعداد واحد آنها:

```
SELECT CourseName, Unit
FROM course
```

- مشاهده فهرستی از نام و نام خانوادگی هنرآموزان به همراه شماره تلفن آنها:

```
SELECT teacherName, teacherFamily, teacherTel
FROM teacher
```

- مشاهده دروس ثبت نام شده:

```
SELECT teach.CourseCode, CourseName
FROM teach, course
WHERE teach.CourseCode = course.CourseCode
```

برای جلوگیری از نمایش مقادیر تکراری یک ستون در پرس و جو

برای این کار به همراه دستور SELECT دستور DISTINCT را اضافه می کنیم.

```
SELECT DISTINCT teach.CourseCode, CourseName
FROM teach, course
WHERE teach.CourseCode = course.CourseCode
```

برای تشخیص اینکه فیلد متعلق به کدام جدول است، از شکل «نام فیلد. نام جدول» استفاده می شود.

یادداشت



برای نمایش ستون ها و جدول ها با نام موقت و یا مستعار از دستور زیر استفاده می کنیم

```
SELECT StudentCode, StudentName + ' ' + StudentFamily AS هنرجو
FROM student
```

شکل کلی دستور SELECT برای مرتب سازی

```
SELECT field_list
FROM table_name
[WHERE selectcriteria]
[ORDER BY field1[ASC | DESC][,field2 [ASC | DESC]][,...]]
```

در شکل کلی دستورات SQL، علامت | به معنی "یا" است و هنگامی که بین دو مقدار مختلف قرار می گیرد، می توان یکی از آن دو را انتخاب کرد و نباید از هر دو مقدار در یک دستور استفاده کرد. از تاپ علامت های [] و |، اجتناب کنید.

یادداشت



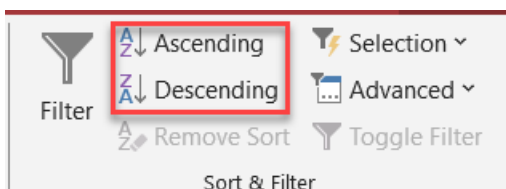
جدول ۷- بخش های مختلف SELECT به همراه ORDER BY

بخش	توضیحات
field_list	نام فیلد یا فیلدهایی که قرار است بازایی شوند. در این قسمت می توان نام مستعار ستون ها را نیز مشخص کرد.
table_name	نام جدولی که قرار است فیلدها از آن بازایی شوند.
selectcriteria	مشخص کننده معیار انتخاب است. در صورتی که از WHERE استفاده شود، Access پس از اعمال معیار جست و جو برای رکوردها، آنها را مرتب می کند.
field1, field2	نام فیلدهایی که مرتب سازی بر اساس آنها انجام خواهد شد.

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

کارگاه ۶: مرتب سازی رکوردها

برای این کار بعد از انتخاب فیلد ها از زبانه Home گروه Sort & Filter روی گزینه Ascending یا Descending کلیک کنید.



کارگاه ۷: گروه بندی نتایج پرس و جو و استفاده از توابع جمعی:

شکل کلی دستور SELECT به همراه GROUP BY

```
SELECT field_list
FROM table_name
WHERE criteria
[GROUP BY groupfieldlist]
```

جدول ۸- بخش‌های مختلف دستور SELECT به همراه GROUP BY

بخش	توضیحات
field_list	نام فیلد یا فیلدهایی که قرار است بازایی شوند. در این قسمت می‌توان نام مستعار ستون‌ها را نیز مشخص کرد. علاوه بر آن از توابع تجمعی می‌توان در این قسمت استفاده کرد.
table_name	نام جدولی که قرار است فیلدها از آن بازایی شوند.
criteria	مشخص‌کننده معیار انتخاب است. در صورتی که از WHERE استفاده شود، Access پس از اعمال معیار جست‌وجو برای رکوردها، آنها را گروه‌بندی می‌نماید.
groupfieldlist	برای گروه‌بندی رکوردها می‌توان نام حداکثر ۱۰ فیلد را در این قسمت مشخص کرد. ترتیب نام فیلدها در این قسمت، مشخص‌کننده سطوح گروه‌بندی از بالاترین به پایین‌ترین سطح است.

- تمامی فیلدهای که در مقابل واژه SELECT وجود دارد حتما باید در GROUP BY قرار گیرند.

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

- برای محاسبه مجموع واحدهای اخذ شده از تابع جمعی SUM استفاده می‌کنیم

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

به این مثال ها توجه کنید:

● نمایش تعداد هنرجویان در هر پایه تحصیلی و گروه‌بندی آنها بر اساس پایه تحصیلی

```
SELECT COUNT(StudentCode), StudentGrade
FROM student
GROUP BY StudentGrade
```

● مشاهده تعداد هنرجویانی که دارای نمره ۱۵ یا بالاتر هستند و گروه‌بندی آنها بر اساس پایه تحصیلی:

```
SELECT StudentGrade, COUNT(student.StudentCode) AS [تعداد هنرجویان]
FROM student, teach
WHERE teach.StudentCode = student.StudentCode AND Mark >= 15
GROUP BY StudentGrade
```

● محاسبه معدل هنرجو و گروه‌بندی آنها بر اساس سال تحصیلی

برای محاسبه معدل، ابتدا مجموع حاصل ضرب نمره در تعداد واحد هر درس را محاسبه کرده (SUM) و سپس بر مجموع واحدها تقسیم می‌کنیم.

```
SELECT student.StudentCode , StudentName , StudentFamily ,
SUM(Mark*Unit) /SUM(Unit) AS [معدل] , EducationYear
FROM student, course, teach
WHERE teach.StudentCode = student.StudentCode AND teach.CourseCode =
course.CourseCode
GROUP BY EducationYear, student.StudentCode, StudentName, StudentFamily
```

● مشاهده فهرستی از نام دروس به همراه تعداد هنرجویانی که در آن درس نمره کمتر از ۱۰ کسب کرده‌اند و گروه‌بندی آنها بر اساس سال و نام درس

```
SELECT EducationYear, CourseName, COUNT(StudentCode) AS [تعداد هنرجویان]
FROM course, teach
WHERE course.CourseCode = teach.CourseCode AND Mark < 10
GROUP BY EducationYear, CourseName
```


کارگاه: ۸ ایجاد گزارش با استفاده از Wizard

مهم ترین نما ها برای نمایش گزارش ها Access :

- نمای layout:
در این نما داده های موجود با شباهت بسیار زیادی به آنچه چاپ می شود وجود دارد و می شود تغییراتی را اعمال کرد. انجام هرگونه تغییراتی که در ظاهر و خوانایی گزارش تأثیر دارد نیز در این نما امکان پذیر است.
- نمای Design:
این نما برای طراحی گزارش استفاده می شود.
این نما جزئیات زیادی برای ویرایش در اختیار کاربر قرار می دهد، در این نما گزارش در حالت اجرایی قرار نمی گیرد و نمی توان در هنگام کار با آن داده ها را مشاهده کرد.
- Print preview:
در این نما آنچه روی کاغذ چاپ می شود نمایش داده می شود و بعد از دیدن و تایید برگه می توان آن را چاپ کرد.

برخی روشهای مختلف برای ایجاد گزارش در Access

- Report: ایجاد یک گزارش ساده
- Report Desing: ایجاد یک گزارش خالی و دادن اجازه به کاربر برای ایجاد فیلدها، اطلاعات و کنترل ها
- Report Wizard: ایجاد گزارش از طریق wizard

ایجاد Wizard

برای ایجاد Wizard از زبانه Create گروه Report گزینه Report Wizard کلیک می کنیم.

کارگاه ۹: ویرایش گزارش

برای ویرایش گزارش از نمای Desing استفاده می‌کنیم.

برای مشاهده گزارش در این نما، روی برگه عنوان گزارش، راست کلیک کرده و گزینه Desing view را انتخاب می‌کنیم

- Report header: در بالای صفحه اول نمایش داده می‌شود و عنوان گزارش را مشخص میکند.
- Page header: در بالای هر صفحه قرار می‌گیرد و برای نمایش سر صفحه‌ها به کار می‌رود.
- Page Footer: در پایین هر صفحه قرار می‌گیرد و شماره صفحه و تعداد کل صفحات را نمایش می‌دهد.
- Detail: بین Header Page و Footer Page قرار می‌گیرد و رکوردهای جدول یا پرس‌وجو را نمایش می‌دهد.
- Report footer: این بخش اختیاری است. در آخرین صفحه گزارش قرار می‌گیرد و خالصه اطلاعات را نمایش می‌دهد.