



SQL

for Data Scientists



OnAcademy24.com

در این دوره، شما به عنوان یک دانشمند داده :

۱. با مفاهیم پایه در پایگاه داده‌های رابطه‌ای آشنا می‌شوید.
۲. بازیابی داده از پایگاه داده را فرا می‌گیرید.
۳. با نحوه نوشتن کوئری‌های ساده و پیچیده مختلف آشنا می‌شوید.
۴. با مثال‌های متنوع بازیابی اطلاعات از یک جدول و چندین جدول را یاد می‌گیرید.
۵. کار کردن با برنامه مدیریت پایگاه داده SQLite را می‌آموزید.

پایگاه داده چیست؟

پایگاه داده یک مجموعه از داده‌های مرتبط با یکدیگر است.

مثال‌های روزمره: دفترچه تلفن

لیست خرید

لیست کارهای روزانه





یک پایگاه داده میتواند به روش‌های مختلفی ذخیره بشود.

روی کاغذ

در ذهن شما

در یک کامپیوتر

مقایسه

لیست خرید کاغذی

لیستی از محصولاتی که باید بخرید

تعداد ۱۰ تا ۲۰ رکورد به طور معمول

صرفا استفاده شخصی دارد.

امنیت داده‌ها خیلی مطرح نیست

سایت دیجی کالا

لیستی از محصولات، قیمت،
مقایسه و بررسی کالا، نظرات
کاربران، امتیاز کالا

تعداد رکوردهای میلیونی

اطلاعات ثبت شده ارزش زیادی
برای سایت دیجی کالا دارد

امنیت داده‌ها بسیار اهمیت دارد

یک ادمین پایگاه داده

دسترسی‌های مجاز کاربران را تعریف می‌کند.

ایجاد جداول مختلف در پایگاه داده

نحوه دسترسی به اطلاعات را تعریف می‌کند.

طراحی و مدیریت کل پایگاه داده

یک دانشمند داده

کاربر نهایی یک پایگاه داده است

استفاده از دستورات SQL

بازیابی اطلاعات مورد نیاز با کوئری زدن روی جداول

درگیر طراحی پایگاه داده نمی‌شود.

01

مشکلات سیستم‌های سنتی
ذخیره و بازیابی داده‌ها

افزایش افزونگی داده

چون در سیستم ذخیره و بازیابی اطلاعات سنتی، داده‌ها به مرور جمع‌آوری میشوند. بنابراین، ممکن است داده‌ها در فایل‌ها پخش شوند و در نتیجه در فایل‌های مختلف تکرار گردند.

یکی از ویژگی‌های بانک‌های اطلاعاتی به اشتراک گذاری و اجتماع داده‌ها است. به اشتراک گذاری و اجتماع داده‌ها میتواند باعث کاهش افزونگی داده‌ها شود.

افزایش بی‌نظمی و ناسازگاری داده‌ها

از آن جایی که در روش سنتی ذخیره سازی، داده ها در فایل های مجزا ذخیره می‌شوند و ممکن است داده ها در چند فایل تکرار بشوند، تغییر داده‌های تکراری می‌تواند موجب بی‌نظمی و ناسازگاری بشود.

پیچیدگی زیاد فایل‌ها

برای مدیریت فایل‌ها باید برنامه‌هایی نوشته بشوند تا کارهایی مثل ویرایش، حذف و بازیابی رکوردها را انجام دهند. نوشتن چنین برنامه‌هایی با توجه به نوع دستیابی به فایل پیچیده خواهد شد.

افزایش مشکلات جامعیت داده

محیط‌های پردازش فایل سنتی، اجازه تعریف قواعد جامعیت را نمیدهند. به عنوان مثال، تست نمیکنند نمره خارج از بازه ۰ تا ۲۰ نباشد یا بررسی نمیکنند که مانده حسابی منفی نشود.

جامعیت داده یعنی سازگاری، دقت و صحت داده‌های ذخیره شده در پایگاه داده در تمام لحظات.

وجود مشکلات امنیتی

همه کاربران نباید به تمام داده‌ها دسترسی داشته باشند. به عنوان مثال، در یک سیستم دانشگاه، استاد باید فقط به اطلاعات دانشجویان خودش برای همان درس دسترسی داشته باشد. واحد مالی باید به اطلاعات مالی و آموزش باید به اطلاعات آموزشی دسترسی داشته باشد. ایجاد این دسترسی‌ها در سیستم‌های پردازش فایل سنتی دشوار است.

دسترسی کندتر

در سیستم‌های سنتی ممکن است اطلاعات دقیق و بروز بازیابی نشوند. زیرا اطلاعات به صورت مجتمع و یکپارچه نگهداری نمی‌گردند.

نیاز به بودجه کمتر

در نگهداری فایلها به روش دستی و سنتی به فضای زیادی نیاز داریم ولی با ذخیره این اطلاعات در کامپیوتر، دیگه نیازی به ساختمانهای بزرگ برای کارمندان جهت نگهداری و پردازش اطلاعات نیست.

02

DataBase Management Systems (DBMS)

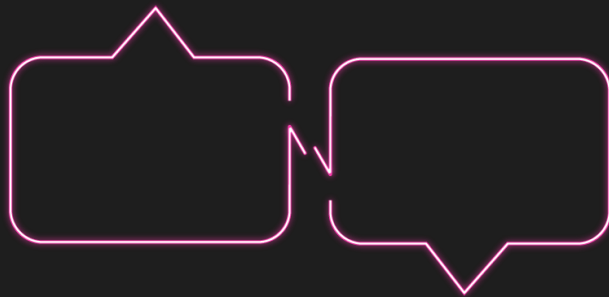
سیستم مدیریت پایگاه داده

برای ذخیره یک پایگاه داده در کامپیوتر میتوان از فایل تکست، فایل اکسل یا... استفاده کرد.

ولی معمولاً از یک برنامه مدیریت پایگاه داده به نام database management system استفاده می‌کنیم.

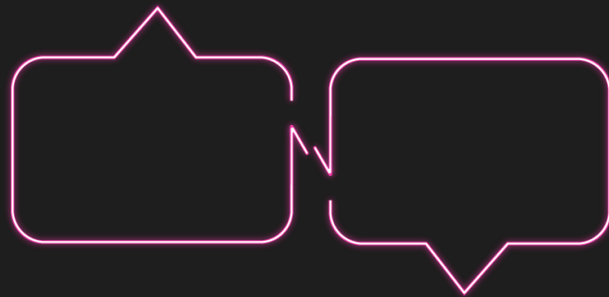
مزایای سیستم‌های مدیریت پایگاه داده

ذخیره و مدیریت حجم زیادی از اطلاعات را راحت می‌کنند.



امنیت داده‌ها رو می‌شود مدیریت کرد.

امکان بکاپ گرفتن منظم از داده‌ها



داده‌های ثابت و قابل اعتماد

هر سیستم از ۳ بخش تشکیل شده است



اطلاعات

داده‌ای که روی آن پردازش انجام شده
و از سیستم خارج شده



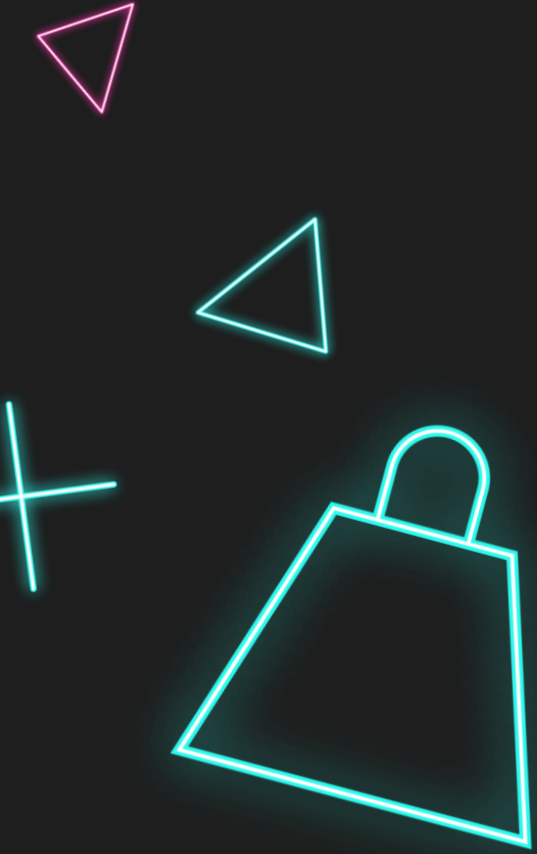
پردازش

فرآیندی که داده را به اطلاعات تبدیل میکند



داده

هر چیزی که وارد سیستم می‌شود



یک سیستم مدیریت پایگاه داده به شما اجازه ۴ عملیات اصلی CRUD را می‌دهد.

C.

Create

R.

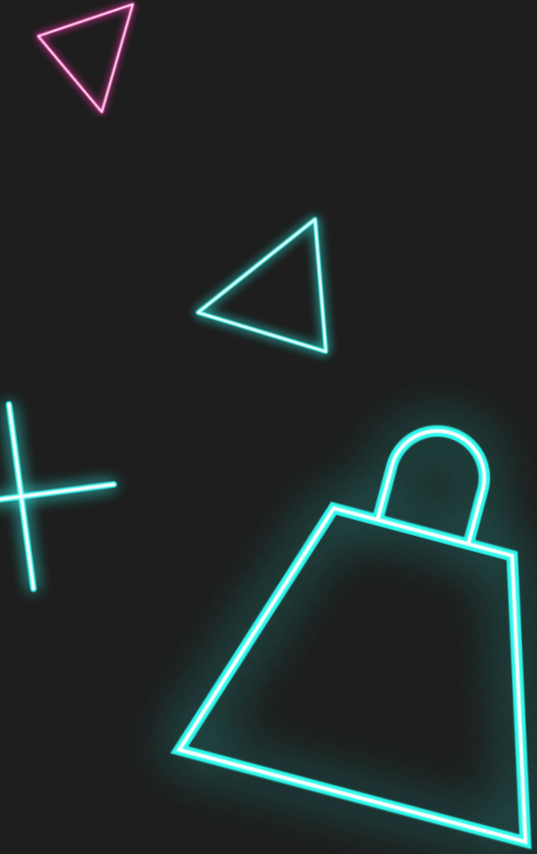
Read

U.

Update

D.

Delete



ما به یک سیستم مدیریت پایگاه داده کوئری می‌زنیم تا یک سری اطلاعات خاص را دریافت کنیم. حالا هرچه که ساختار یا SCHEMA ی یک پایگاه داده پیچیده‌تر و تعداد جداول پایگاه داده بیشتر باشد، بازیابی اطلاعات از جداول دشوارتر است.

سوال: چند نمونه کوئری؟

انواع پایگاه داده‌ها

۱. پایگاه داده‌های رابطه‌ای

اطلاعات آن‌ها در یک یا چندین پایگاه جدول ذخیره شده است.

هر جدول سطرها و ستون‌های مختلفی دارد که هر سطر شماره منحصر به فردی دارد.

شما می‌توانید به صورت سطر به سطر به آن اطلاعات اضافه کنید.

Relational Databases (SQL)

Student Table

*ID #	Name	Major
1	Jack	Biology
2	Kate	Sociology
3	Claire	English
4	John	Chemistry

Users Table

*Username	Password	Email
jsmith22	wordpass	...
catlover45	apple223	...
gamerkid
giraffe

۱. پایگاه داده‌های رابطه‌ای

برنامه‌های مدیریت پایگاه داده رابطه‌ای به کاربرها اجازه میدهند که یک پایگاه داده رابطه‌ای را ساخته و از آن استفاده کنند.

چند برنامه معروف:

my sql

oracle

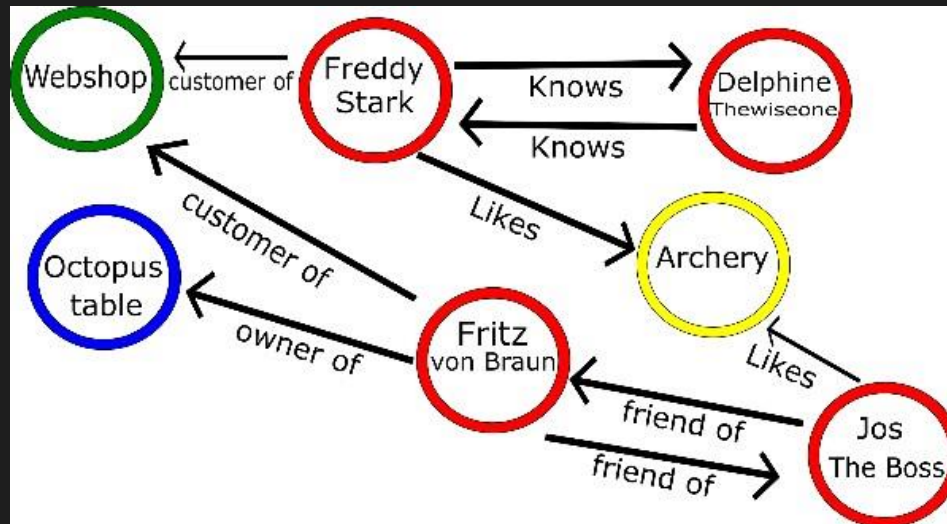
sqlite

sql server

۱. پایگاه داده‌های غیر رابطه‌ای

هر پایگاه داده‌ای که رابطه‌ای نباشد، غیر رابطه‌ای محسوب می‌شود.

هر زمان که نتوانیم داده را به طور ساختاریافته در جداول یک پایگاه داده رابطه‌ای قرار دهیم از این پایگاه داده‌ها استفاده می‌کنیم.



خلاصه مطالبی که گفته شد



یک پایگاه داده یک مجموعه از داده‌های مرتبط با هم است.

کامپیوترها برای ذخیره پایگاه داده‌ها بسیار مناسب هستند.

سیستم‌های مدیریت پایگاه داده ایجاد، نگهداری و امنیت یک پایگاه داده را راحت‌تر می‌کنند.

سیستم‌های مدیریت پایگاه داده، به شما اجازه می‌دهند تا عملیات **CRUD** و سایر عملیات مدیریتی را اجرا کنید.

به طور کلی ۲ نوع پایگاه داده رابطه‌ای و غیر رابطه‌ای داریم.

پایگاه داده‌های رابطه‌ای، از زبان **SQL** استفاده می‌کنند و داده‌ها را در جداول در سطرها و ستون‌ها ذخیره می‌کنند. پایگاه داده‌های غیر رابطه‌ای از ساختارهای دیگری برای ذخیره داده‌ها استفاده می‌کنند.

03

مدل رابطه‌ای

مدل رابطه‌ای

در مدل رابطه‌ای، داده‌ها به صورت جدول نمایش داده می‌شود.
مدل رابطه‌ای در سال ۱۹۷۰ توسط ریاضیدانی به اسم cod طراحی شد.
گفتیم که یک پایگاه داده رابطه‌ای، از تعدادی جدول تشکیل میشود.
مفاهیم اولیه مدل رابطه‌ای:

دامنه

تاپل

درجه رابطه

کاردینالیتی رابطه

کلید

مدل رابطه‌ای

دامنه: مجموعه مقادیری که هر ویژگی (ستون) میتواند بگیرد. مثل عنوان مدارک تحصیلی رابطه، حاصل ضرب دکارتی چندین دامنه است.

```
DRanking = { فوق لیسانس، لیسانس و فوق دیپلم }  
DLname = { خداینده، عباس نژاد، جلالی، حسینی و ابوطالبی }
```

مدل رابطه‌ای

تاپل هر زوج مرتب از رابطه را یک تاپل می‌گویند. به زبان ساده‌تر، هر سطر یا رکورد جدول یک تاپل نام دارد. یک نمونه تاپل در جدول مقابل : (جلالی، دکتری)

DLname	DRanking
ابوطالبی	دکتری
حسینی	فوق لیسانس
جلالی	دکتری
هاشمی	فوق لیسانس
عباس نژاد	فوق لیسانس
خداپنده	فوق لیسانس

ضرب دکارتی دو دامنه، دوتایی‌های مرتب ایجاد می‌کند.

ضرب دکارتی سه دامنه، سه‌تایی‌های مرتب ایجاد می‌کند.

ضرب دکارتی چند دامنه، چندتایی‌های مرتب را ایجاد می‌کند.

مدل رابطه‌ای

درجه رابطه: تعداد صفات رابطه (ستون‌های جدول) درجه رابطه را به شما نشان می‌دهد.

مثلا اگر جدول شما از ۵ صفت کد استادی، نام استاد، نام خانوادگی استاد، مدرک تحصیلی، شماره تماس استاد تشکیل شده باشد. درجه این رابطه برابر با ۵ خواهد بود.

مدل رابطه‌ای

کاردینالیتی رابطه: تعداد تاپل‌های رابطه (تعداد رکوردها یا ردیف‌های جدول). را کاردینالیتی رابطه می‌گویند

مفاهیم ریاضی یک رابطه و معادل آن‌ها در یک جدول				
مدل رابطه‌ای	رابطه	تاپل	صفت	دامنه
کاربر	جدول	ردیف	ستون	مقادیری که هر ستون می‌پذیرد.
برنامه نویس	فایل	رکورد	فیلد	مقادیری که هر فیلد میتواند بپذیرد.

مدل رابطه‌ای

کلید: شناسه‌ای در یک رابطه است که منحصر به فردی رکوردهای (تاپل‌های) رابطه را تضمین میکند. یعنی، این شناسه در یک رابطه مقدار تکراری نمی‌پذیرد.

کلید میتواند یک صفت یا ترکیبی از چند صفت باشد.

برخی از کلیدهایی که میتوانید تعریف کنید عبارتند از:

شماره دانشجویی برای هر دانشجو هر دانشجو یک شماره دانشجویی منحصر به فرد دارد و شماره دانشجویی دو دانشجو یکسان نیست.

کد کارمندی برای کارمندان به هر کارمند فقط یک کد کارمندی یکتا تخصیص داده می‌شود.

کد ملی برای اشخاص به هر فردی یک کد ملی منحصر به فرد تخصیص می‌یابد و افراد نمی‌توانند کد ملی یکسانی داشته باشند.

مدل رابطه‌ای

دلایلی برای استفاده از انواع کلید در یک سیستم مدیریت پایگاه داده:

کلیدها به شناسایی هر سطری از داده‌ها در جدول کمک می‌کنند..

کلید در پایگاه داده این امکان را فراهم می‌آورد که روابط بین جداول را ایجاد و همچنین، روابط بین جداول را شناسایی کرد.


کلید به ایجاد هویت و یکپارچگی در روابط میان جدول‌ها در پایگاه داده کمک می‌کند.

مدل رابطه‌ای

کلید در رابطه دارای ویژگی‌های زیر است:

مقدار کلید نمیتواند تکراری باشد.

مقدار کلید نمیتواند تهی باشد.

 سوال: آیا میتوان برای یک رابطه کلیدی تعریف نکرد؟

مدل رابطه‌ای

کلیدها در پایگاه داده انواع مختلفی دارند و هر کلید کارکرد متفاوت خودش را دارد.

در ادامه، تعریف انواع کلید در پایگاه داده را به همراه کارکرد اختصاصی آن‌ها یاد می‌گیریم.

Super Key

Roll_No	Name	Age	Phone
1	Arya	21	7491901521
2	Bran	19	8491901000
3	John	24	9291018403
4	Max	24	7903084562

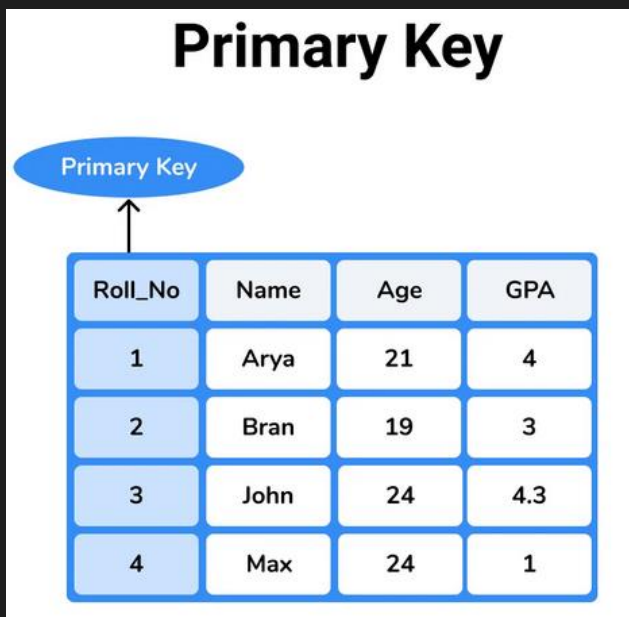
سوپر کلید یا Super Key یک گروه از ستون‌های یکتا یا چندگانه است

که سطرها را در جدول شناسایی می‌کنند.

مدل رابطه‌ای

کلید اصلی یا Primary Key یک ستون یا گروهی از ستون‌ها در جدول است که به طور یکتا هر سطر در آن جدول را شناسایی می‌کند.

Primary Key



The diagram illustrates a database table with four columns: Roll_No, Name, Age, and GPA. The Roll_No column is highlighted with a blue background and is labeled as the Primary Key with an arrow pointing to it from a blue oval containing the text 'Primary Key'.

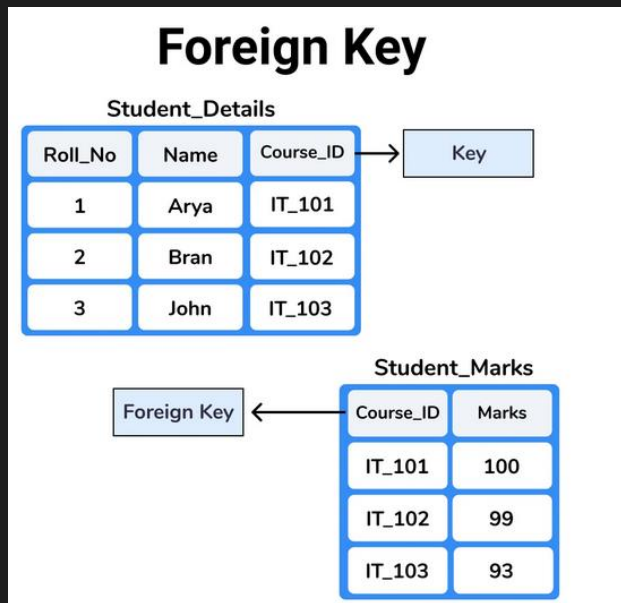
Roll_No	Name	Age	GPA
1	Arya	21	4
2	Bran	19	3
3	John	24	4.3
4	Max	24	1

- مقادیر ستون کلید اصلی باید مقادیر منحصر به فردی داشته باشند.
- نمیتوانید برای ستونی که به عنوان کلید اصلی تعریف شده، مقدار null قرار بدید.
- هر جدول میتواند فقط یک کلید اصلی داشته باشد. البته می‌توانید ترکیبی از چند ستون را به عنوان کلید اصلی تعریف کنید.

مدل رابطه‌ای

کلید خصوصی یا Foreign Key کلیدی است که رابطه بین دو جدول را می‌سازد. هدف از کلید خارجی آن است که یکپارچگی داده‌ها حفظ شود.

- کلید خارجی به یک کلید اصلی در جدولی دیگر اشاره می‌کند.
- نوع و تعریف ستون‌ها در هر دو جدول باید مشابه همدیگر باشد.



مدل رابطه‌ای

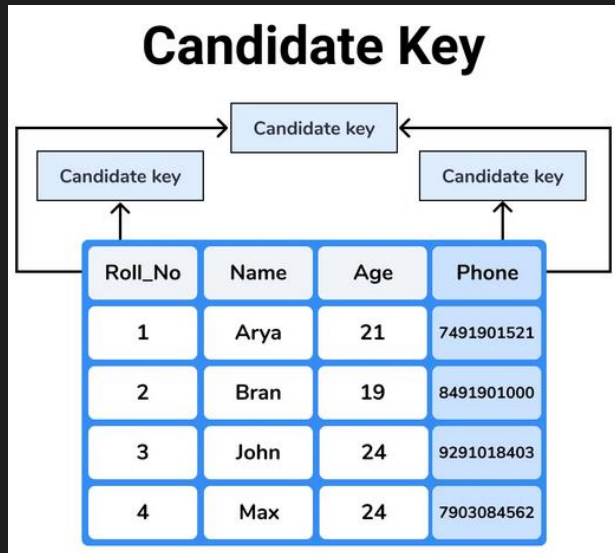
کلید کاندید یا Candidate Key یک مجموعه از ستون‌هاست که به طور یکتا، ردیف‌های آن جدول را شناسایی می‌کند.

- مقادیر ستون کلید کاندید نمیتوانند null باشد.

- مقادیر ستون کلید کاندید باید یونیک باشد.

- به کلید کاندید مینی سوپر کلید هم میگویند. چون معمولا

کلیدهای کاندید از دل سوپر کلیدها می‌آیند.



مدل رابطه‌ای

کلید جایگزین یا alternate key ستونی از جدول است که قابلیت انتخاب به عنوان کلید اصلی را دارد ولی به هر دلیلی به عنوان کلید اصلی انتخاب نشده است

Alternate Key

Roll_No	Name	Age	Phone
1	Arya	21	7491901521
2	Bran	19	8491901000
3	John	24	9291018403
4	Max	24	7903084562

04

دستورات SQL

دستورات CRUD

- دستور CREATE برای ایجاد جدول جدید در پایگاه داده. از دستور INSERT نیز برای افزودن داده‌ها به پایگاه داده استفاده می‌شود.
- دستور UPDATE برای بروزرسانی رکوردهای فعلی جدول
- دستور DELETE برای حذف رکوردهای جدول
- دستورات اسلایدهای بعدی برای خواندن و بازیابی اطلاعات به کار می‌روند.

دستور SELECT

- دستور SELECT برای انتخاب استفاده می‌شود و یکی از ساده‌ترین دستورات SQL و پایه‌ترین دستوری است که تقریباً در همه دستوره‌های دیگر هم از آن استفاده می‌شود.
- بهتر است کلمات کلیدی SQL را به صورت حروف بزرگ بنویسید تا خواندن و درک دستورها آسان‌تر شود.
- همان‌طور که از نام این دستور مشخص است، از آن برای انتخاب داده‌ها از یک جدول استفاده می‌شود.

دستور WHERE

- دستور SELECT برای بازیابی داده‌ها مناسب است ولی اگر بخواهید موارد انتخابی را کمی فیلتر کنید، می‌توانید از دستور WHERE استفاده کنید.
- این دستور امکان اعمال شروطی را برای دستور SELECT را فراهم می‌کند و می‌توان آن‌ها را به انتهای آن دستور اضافه کرد.

دستور ORDER

- دستور ORDER برای مرتب‌سازی نتایج باز یابی شده استفاده می‌شود.
 - برای استفاده از این دستور نیز کافی است آن را به انتهای دستور خود اضافه کنید.
- در این دستور باید نام ستون و نوع ترتیب را مشخص کنید. همچنین می‌توانید بر اساس چند ستون، مرتب‌سازی را انجام دهید.

ASC برای ترتیب صعودی

DESC برای ترتیب نزولی

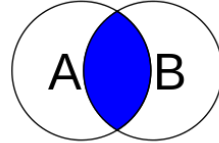
دستور GROUP BY

- با استفاده از دستور GROUP BY می توان دسته بندی یا گروه بندی ستون ها را انجام داد .
- گروه بندی را بر اساس ستونی که کلید اصلی است قرار نمی دهیم، چون هیچ یک از فیلدهای آن تکراری نیست. (در واقع گروه بندی را روی ستون هایی که دارای فیلد تکراری هستند انجام می دهیم.)

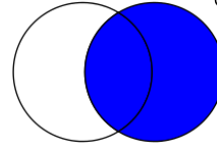
دستور JOIN

- دستور **JOIN** برای الحاق داده‌های مرتبط، که در چند جدول ذخیره شده‌اند، استفاده می‌شود.
- با استفاده از این دستور، جدول دوم به جدول اول ملحق می‌شود و تعیین می‌شود که داده‌ها چگونه با هم ارتباط دارند.
- عبارت **ON** طور صریح مشخص می‌کند که کدام ستون‌ها باید به عنوان کلید استفاده شوند.

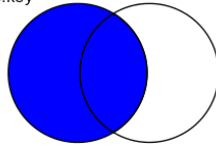
SELECT <fields>
FROM TableA A
INNER JOIN TableB B
ON A.key = B.key



SELECT <fields>
FROM TableA A
RIGHT JOIN TableB B
ON A.key = B.key

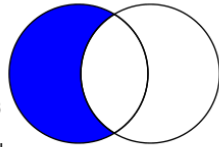


SELECT <fields>
FROM TableA A
LEFT JOIN TableB B
ON A.key = B.key

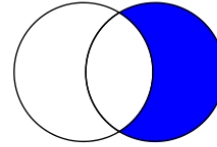


SQL JOINS

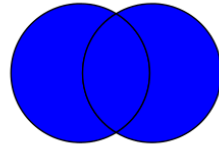
SELECT <fields>
FROM TableA A
LEFT JOIN TableB B
ON A.key = B.key
WHERE B.key IS NULL



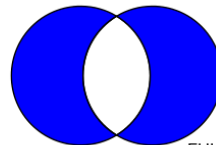
SELECT <fields>
FROM TableA A
RIGHT JOIN TableB B
ON A.key = B.key
WHERE A.key IS NULL



SELECT <fields>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.key = B.key



SELECT <fields>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.key = B.key
WHERE A.key IS NULL
OR B.key IS NULL



دستور JOIN

(INNER) JOIN ردیف‌هایی را که در هر دو جدول انطباق داشته باشند، بازمی‌گرداند.

LEFT (OUTER) JOIN همه ردیف‌های جدول چپ را بازمی‌گرداند و ردیف‌هایی از جدول راست را که انطباق داشته باشند نیز بازمی‌گرداند. اگر هیچ انطباقی وجود نداشته باشد، رکوردهای جدول چپ همچنان بازگردانی می‌شوند.

دستور JOIN

RIGHT (OUTER) JOIN این دستور مخالف دستور الحاق از چپ است، همه ردیفهای جدول راست بازگردانی می شوند و ردیفهایی از جدول راست که انطباق داشته باشند نیز بازیابی می شوند.

FULL (OUTER) JOIN هر رکوردی را که در هر کدام از جدول ها انطباق داشته باشد، بازیابی می کند.

خواص فیلدها در جدول:

هر فیلد در جدول یک نام یکتا دارد
نوع هر فیلد باید مشخص شود
محدودیت‌ها یا قیدهای یک فیلد باید مشخص شود

DEFAULT محدودیت

UNIQUE محدودیت

NOT NULL محدودیت

CHECK محدودیت

تمرین بیشتر

با توجه به اطلاعات روبرو کوئری‌های خواسته شده در اسلاید بعد را بنویسید.

نام کارمند	<u>employee_name</u>	Employee
خیابان	street	
شهر	city	
نام کارمند	<u>employee_name</u>	Works
نام شرکت	<u>company_name</u>	
حقوق	salary	
نام شرکت	<u>company_name</u>	Company
شهر محل شرکت	city	
نام کارمند	<u>employee_name</u>	Manages
نام مدیر	<u>manager_name</u>	

تمرین بیشتر

1. کارمندانی را بازیابی کنید که برای شرکت مخابرات تهران کار می کنند
2. کوئری بنویسید که نام، خیابان و شهرهای محل زندگی کارمندانی را بازیابی کند که برای شرکت مخابرات تهران کار می کنند و دریافتی آن ها بیشتر از ۱۰ میلیون تومان است.
3. کارمندانی را بازیابی کنید که از تمام کارمندان مخابرات تهران بیشتر حقوق می گیرند.

تمرین بیشتر

با توجه به جدول زیر، یک کوئری بنویسید که مشتریانی را که وام نگرفته اند، مشخص کند

Customer (<u>custName</u> , address)	مشتری
Account (<u>accNo</u> , custName)	حساب
Loan (<u>loanNo</u> , custName)	وام

کوئری‌هایی که می‌خواهیم روی دیتابیس **movies** بزنیم:

- یک کوئری **SQL** بنویسید که عناوین تمام فیلم‌های منتشر شده در سال ۲۰۰۸ را فهرست کند.
- یک کوئری بنویسید که سال تولد **Emma Stone** را بنویسد. فرض کنید که تنها یک نفر در پایگاه داده با نام **Emma Stone** وجود دارد.
- یک کوئری **SQL** بنویسید تا عناوین همه فیلم‌های با تاریخ انتشار ۲۰۱۸ یا بعد از آن را به ترتیب حروف الفبا فهرست کند.
- یک کوئری **SQL** بنویسید تا تعداد فیلم‌های با رتبه **IMDb 10.0** مشخص شود.

کوئری‌هایی که می‌خواهیم روی دیتابیس **movies** بزنیم:

- یک کوئری **SQL** بنویسید تا عناوین و سال‌های انتشار همه فیلم‌های هری پاتر را به ترتیب زمانی فهرست کند.
- یک کوئری **SQL** بنویسید و میانگین امتیاز همه فیلم‌های منتشر شده در سال ۲۰۱۲ را تعیین کنید.
- یک کوئری **SQL** بنویسید و تمام فیلم‌های منتشر شده در سال ۲۰۱۰ و رتبه‌بندی آنها را به ترتیب نزولی بر اساس رتبه‌بندی فهرست کنید. برای فیلم‌هایی با رتبه یکسان، آنها را بر اساس حروف الفبا مرتب کنید.
- یک کوئری **SQL** بنویسید تا نام تمام افرادی که در **Toy Story** بازی کرده‌اند، فهرست شود.

کوئری‌هایی که می‌خواهیم روی دیتابیس movies بزنیم:

○ یک کوئری SQL بنویسید که نام تمام افرادی را که در فیلمی که در سال ۲۰۰۴ منتشر شده بازی کرده اند، به ترتیب بر اساس سال تولد، فهرست کند.

۱. کوئری شما باید یک جدول با یک ستون برای نام هر فرد خروجی دهد.

۲. افراد دارای سال تولد یکسان می‌توانند به هر ترتیبی فهرست شوند.

۳. نیازی نیست نگران افرادی باشید که سال تولدشان در لیست ذکر نشده است، تا زمانی که کسانی

که سال تولد دارند به ترتیب ذکر شده باشند.

۴. نام هر شخص تنها یک بار در لیست نمایش داده شود.

کوئری‌هایی که می‌خواهیم روی دیتابیس **movies** بزنیم:

○ یک کوئری **SQL** بنویسید تا نام تمام افرادی که فیلمی را کارگردانی کرده‌اند که حداقل امتیاز ۹.۰ را دریافت کرده‌اند، فهرست کند.

۱. کوئری شما باید یک جدول با یک ستون برای نام هر فرد خروجی دهد.

۲. اگر شخصی بیش از یک فیلم را کارگردانی کرده است که آن فیلم حداقل امتیاز ۹.۰ را دریافت کرده است، باید فقط یک بار در نتایج شما ظاهر شود.

○ یک کوئری **SQL** بنویسید تا عناوین پنج فیلم با بالاترین امتیاز (به ترتیب) که **Boseman Chadwick** در آنها بازی کرده است، فهرست شود. لیست با بالاترین امتیاز شروع شود.

کوئری‌هایی که می‌خواهیم روی دیتابیس **movies** بزنیم:

○ یک کوئری SQL بنویسید تا عناوین تمام فیلم‌هایی را که **Johnny Depp** و **Helena Bonham Carter** در

آن‌ها بازی کرده‌اند، فهرست کند.

۱. کوئری شما باید یک جدول با یک ستون برای عنوان هر فیلم خروجی دهد.

۲. فرض کنید که تنها یک نفر در پایگاه داده با نام **Johnny Depp** وجود دارد.

۳. فرض کنید که تنها یک نفر در پایگاه داده با نام **Helena Bonham Carter** وجود دارد.