# برنامه نویسی دستگاه های سیار (CE364)

## جلسه پانزدهم:
## مقدمه ای بر پایگاه داده رابطه ای

**سجاد شیرعلی شهرضا**

**پاییز 1401**

***چهارشنبه، 30 آذر 1401***

# اطلاع رسانی

- بخش مرتبط با این جلسه:

- Unit 5: Data persistence:
  - Pathway 1: Introduction to SQL, Room, and Flow


# سوال؟

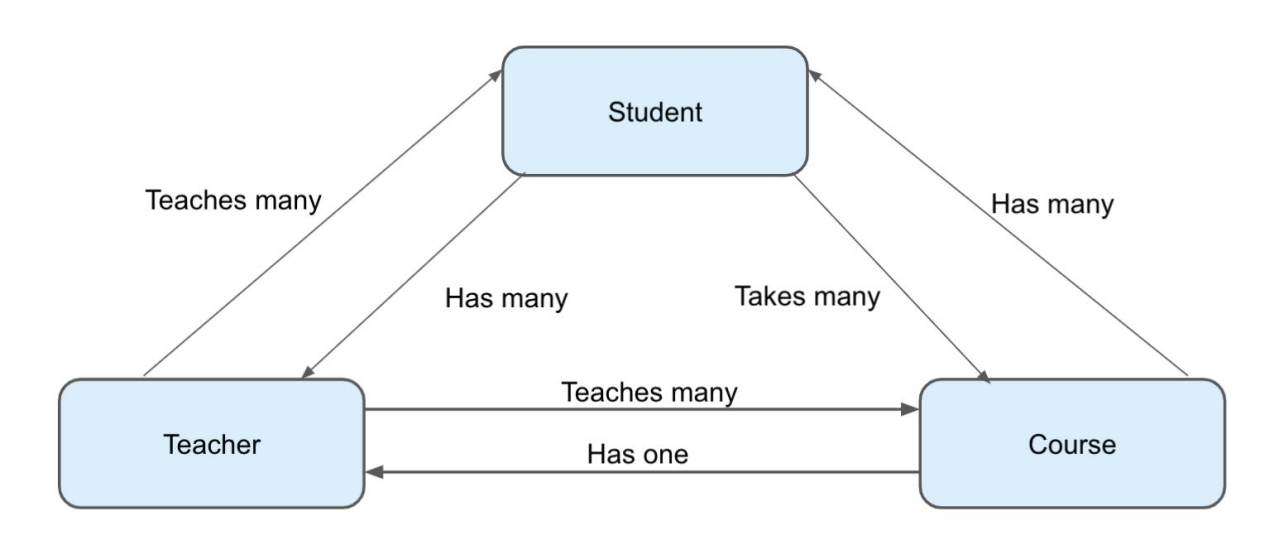
# مقدمه

# پایگاه داده

- یک مجموعه ساختاردار از اطلاعات
- کاربرد در برنامه ها: نگه داری اطلاعات برای اجراهای بعدی
  - برطرف کردن نیاز به دریافت دوباره اطلاعات (مثلا از اینترنت)
  - امکان پذیر کردن ادامه کار
  - معروف به ماندگاری داده (data persistence)

# پایگاه داده رابطه ای (relational database)

- ذخیره اطلاعات در قالب مجموعه ای از جدول ها، ستون ها، و سطرها
- امکان ارجاع به اطلاعات در جداول دیگر



# ذخیره اطلاعات در جدول

- جدول: مجموعه ای از اطلاعات مشابه (مثلا دانشجویان)
  - مشابه تعریف یک کلاس
  - دارای یک نام
- ستون: یک ویژگی از هر نمونه داده
  - مشابه یک ویژگی یا مقدار در یک کلاس
  - دارای یک نام و یک نوع
- سطر: یک نمونه از داده
  - مشابه یک شی ایجاد شده از روی کلاس

| Plant | |
|---|---|
| id | INTEGER |
| species | TEXT (string) |
| name | TEXT (string) |
| color | TEXT (string) |

| Garden | |
|---|---|
| id | INTEGER |
| name | TEXT (string) |
| length | INTEGER |
| width | INTEGER |

| id | species | name | color |
|---|---|---|---|
| 1 | Camellia Sinensis | Tea Plant | green |
| 2 | Echinacea Purpurea | Purple Coneflower | purple |
| 3 | Ferula Foetida | Asafoetida | green |

# کلید اصلی (Primary Key)

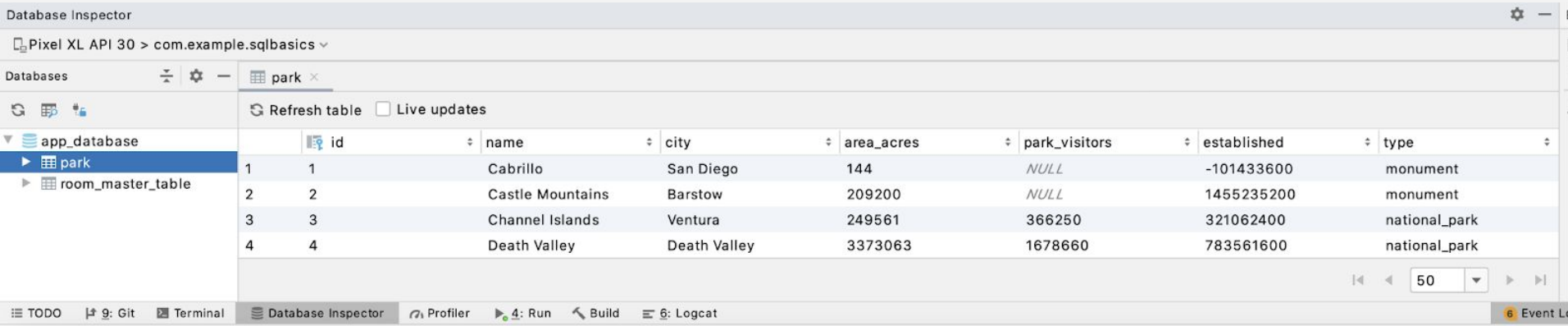
- مقدار یکتا (Unique) برای هر نمونه از داده
- عدم امکان تکرار
- ستون مورد استفاده برای ارجاع به یک سطر خاص
- کاربرد اصلی: ارجاع از جداول دیگر

# تعامل با پایگاه داده رابطه ای

- زبان SQL
  - مخفف: Structured Query Language
- دستورات اصلی:
  - جستجو (SEARCH): پیدا کردن اطلاعات در پایگاه داده با توجه به برخی ویژگی ها
  - اضافه کردن (INSERT): اضافه کردن یک سطر جدید
  - به روز رسانی (UPDATE): تغییر یک و یا چندین سطر
  - حذف (DELETE): حذف یک یا چندین سطر

# محیط آزمایشی

- اجرای نمونه برنامه داده شده
- کارکرد برنامه: ایجاد یک پایگاه داده
- فراهم آمدن امکان مشاهده و کار با پایگاه داده ایجاد شده در اندروید استودیو
- استفاده از ابزار «وارسی پایگاه داده» (Database Inspector)



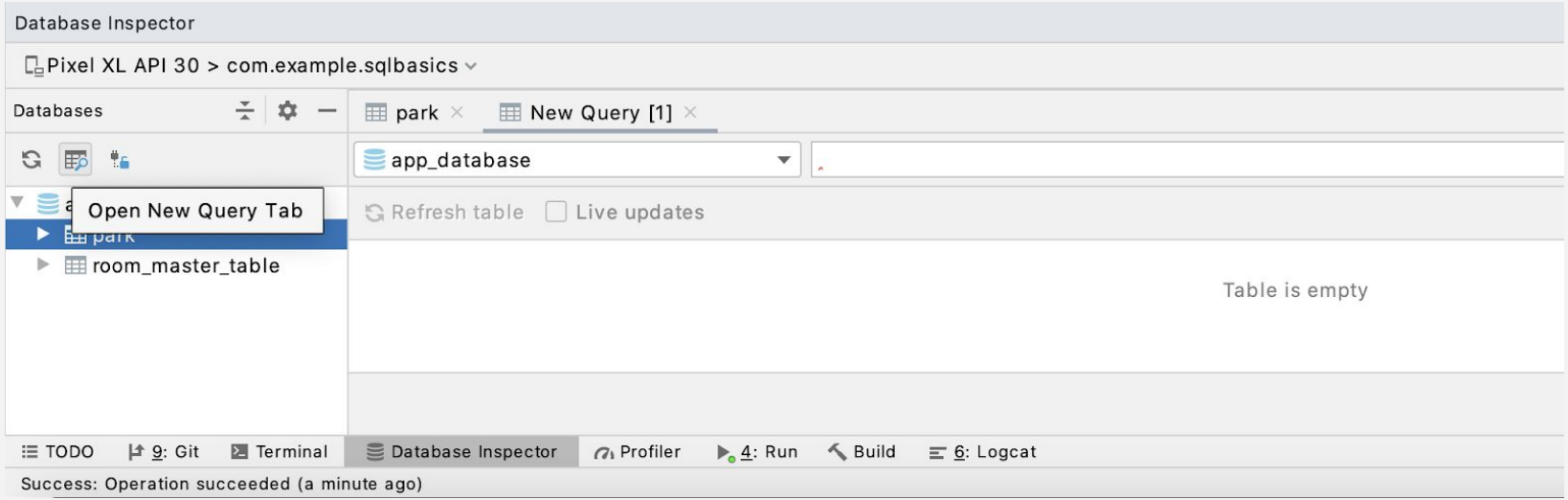

# سوال؟

# انتخاب داده در جدول

**چگونگی استفاده از دستور SELECT**

# استفاده از ابزار وارسی پایگاه داده

- اجرای ابزار
- انتخاب جدول (park در این مثال)
- ایجاد یک برگه جدید برای اجرای دستور



# نمونه های دستور انتخاب (SELECT)

- دریافت کل اطلاعات (همه ستون ها) از یک جدول

```
SELECT * FROM park
```

- دریافت یک ستون خاص

```
SELECT city FROM park
```

- دریافت مجموعه ای از ستون ها

```
SELECT name, established, city FROM park
```

# محدود کردن جواب ها

- محدود کردن تعداد جواب های برگشتی

```
SELECT name FROM park
LIMIT 5
```

- اضافه کردن شرط برای داده های برگشتی

```
SELECT name FROM park
WHERE type = "national_park"
```

```
SELECT name FROM park
WHERE type != "recreation_area"
AND area_acres > 100000
```

# مثال 1

- پیدا کردن اسم تمام پارک ها با کمتر از یک میلیون بازدید کننده

# مثال 1 (جواب)

- پیدا کردن اسم تمام پارک ها با کمتر از یک میلیون بازدید کننده

```
SELECT name FROM park
WHERE park_visitors < 1000000
```

# برخی توابع پر استفاده

- محاسبه تعداد جواب ها

```
SELECT COUNT(*) FROM park
```

- محاسبه مجموع جواب ها

```
SELECT SUM(park_visitors) FROM park
WHERE type = "national_park"
```

- انتخاب کمترین (و یا بیشترین) جواب

```
SELECT MAX(area_acres) FROM park
WHERE type = 'national_park'
```

- انتخاب جواب های غیر تکراری

```
SELECT DISTINCT type FROM park
```

- ترکیب چند مورد با هم

```
SELECT COUNT(DISTINCT type) FROM park
```

# مثال 2

- پیدا کردن تعداد شهرهایی که اطلاعات پارک های آنها موجود است.

# مثال 2 (جواب)

- پیدا کردن تعداد شهرهایی که اطلاعات پارک های آنها موجود است.

```
SELECT COUNT(DISTINCT city) FROM park
```

# مثال 3

- پیدا کردن تعداد افراد بازدید کننده از پارک ها در شهر سانفرانسیسکو

# مثال 3 (جواب)

- پیدا کردن تعداد افراد بازدید کننده از پارک ها در شهر سانفرانسیسکو

```
SELECT SUM(park_visitors) FROM park
WHERE city = "San Francisco"
```

# مرتب کردن جواب ها

- اضافه کردن "ORDER BY"
- مشخص کردن ستون مورد نظر برای مرتب سازی
- صعودی (ASC) یا نزولی (DESC)
  - پیش فرض: صعودی

```
SELECT name FROM park
ORDER BY name
```

```
SELECT name FROM park
ORDER BY name DESC
```

# دسته بندی جواب ها

- دسته بندی جواب ها بر اساس یک و یا چند ستون
- انجام بقیه قسمت های درخواست بر روی هر دسته

```
SELECT type, name FROM park
GROUP BY type
ORDER BY name
```

```
SELECT type, COUNT(*) FROM park
GROUP BY type
ORDER BY type
```

# مثال 4

- لیست 5 پارک پربازدید به صورت نزولی

# مثال 4 (جواب)

- لیست 5 پارک پربازدید به صورت نزولی

```
SELECT name, park_visitors FROM park
ORDER BY park_visitors DESC
LIMIT 5
```


# سوال؟

# درج، به روز رسانی و حذف

# اضافه کردن یک سطر

- تعیین نام جدول
- تعیین مقادیر ستون های مختلف
  - به ترتیب تعریف شده در جدول

```
INSERT INTO table_name
VALUES (column1, column2, ...)
```

```
INSERT INTO park
VALUES (null, 'Googleplex', 'Mountain View', 12, null, 0, '')
```

# به روزرسانی سطرها

- تعیین سطرهایی که باید به روز شوند
- تعیین مقدار جدید برای ستون ها

```
UPDATE table_name
SET column1 = ...,
column2 = ...,
...
WHERE column_name = ...
...
```

```
UPDATE park
SET area_acres = 46,
established = 1088640000,
type = 'office'
WHERE name = 'Googleplex'
```

# حذف یک یا چندین سطر

- تعیین نام جدول
- تعیین سطرهای مشمول
- حذف **تمامی** سطرهای واجد شرایط

```
DELETE FROM table_name
WHERE <column_name> = ...
```

```
DELETE FROM park
WHERE name = 'Googleplex'
```


# سوال؟