

## مقدمه ای بر SQL

▶ SQL یک زبان استاندارد برای دسترسی و دستکاری پایگاه های داده است.

SQL چیست؟

▶ SQL مخفف Language Query Structured یا زبان پرس و جوی ساخت یافته است.

▶ SQL به شما اجازه دسترسی و دستکاری به پایگاه های داده را می دهد.

▶ SQL یک استاندارد ANSI (موسسه استاندارد ملی آمریکا) است

## چه کارهایی می تواند انجام دهد؟

- ▶ SQL یک پرس و جو را در برابر یک پایگاه داده می تواند سریع اجرا کند
- ▶ SQL می تواند داده را از یک پایگاه داده واکنشی کند
- ▶ SQL می تواند رکوردهایی را در پایگاه داده درج کند
- ▶ SQL می تواند رکوردهایی را در پایگاه داده به روز کند
- ▶ SQL می تواند رکوردهای یک پایگاه داده را حذف کند
- ▶ SQL می تواند پایگاه داده های جدیدی بجهود آورد
- ▶ SQL می تواند جدولهای جدیدی را در پایگاه داده بجهود آورد
- ▶ SQL می تواند رویه های ذخیره شده در پایگاه داده بسازد
- ▶ SQL می تواند نماهایی را در پایگاه داده بجهود آورد
- ▶ SQL می تواند برای جداول، رویه های ذخیره شده و نماها مجوز تنظیم کند

## نقل قولها در اطراف فیلدهای متنی

- ▶ SQL نقل قول های تکی در مورد مقادیر متنی بکار می برد (بیشتر سیستمهای پایگاه داده نقل قول های دوتایی می پذیرند). اما نباید مقادیر عددی داخل نقل قول ها ضمیمه شوند.
- ▶ برای مقادیر متنی:

---

This is correct:

```
SELECT * FROM Persons WHERE FirstName='Tove'
```

This is wrong:

```
SELECT * FROM Persons WHERE FirstName=Tove
```

---

- ▶ برای مقادیر عددی:

---

This is correct:

```
SELECT * FROM Persons WHERE Year=1965
```

This is wrong:

```
SELECT * FROM Persons WHERE Year='1965'
```

---

## عملگر IN در SQL

▶ عملگر IN به شما اجازه می دهد مقادیر چندگانه در عبارت WHERE تعیین کنید .

گرامر IN در SQ

```
SELECT column_name(s)
FROM table_name
WHERE column_name IN (value1,value2,...)
```

▶ مثالی از عملگر IN

جدول "Persons"

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

# JOINها در SQL

▶ JOIN ها در SQL برای پرس و جوی داده از دو یا چند جدول، مبنی بر یک رابطه بین برقی ستون ها در این جدول ها استفاده می شود

## ▶ JOIN در SQL

کلمه کلیدی JOIN برای پرس و جوی داده از دو یا چند جدول در یک عبارت، SQL مبنی بر یک رابطه بین برقی ستون ها در این جدول ها استفاده می شود.  
جدول ها در یک پایگاه داده اغلب با کلیدها با یکدیگر مرتبط هستند.  
کلید اصلی (Primary Key) یک ستون (یا ترکیبی از ستون ها) با یک مقدار منمصر برای هر سطر است. هر مقدار کلید اصلی باید در داخل جدول، منمصر به فرد باشد. هدف پسباندن داده ها به یکدیگر، در سراسر جدول ها، جلوگیری از تکرار همه داده ها در هر جدول است

جدول "Persons" را ببینید:

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

▶ توجه کنید که ستون "P\_Id" یک کلید اصلی در جدول "persons" است. به این معنا که دو سطر، مقدار "P\_Id" یکسان نمی توانند داشته باشند. P\_Id دو شخص را متمایز می نماید؛ متی اگر آن ها نام یکسانی داشته باشند.

پس جدول "Order" را داریم :

O_Id	OrderNo	P_Id
1	77895	3
2	44678	3
3	22456	1
4	24562	1
5	34764	15

► توجه کنید که ستون "O-Id" یک کلید اصلی در جدول "Orders" است و ستون "P-Id" را به جدول "Persons" بدون استفاده از نام آنها ارجاع می دهد. توجه کنید که رابطه بین دو جدول بالا، ستون "P-Id" است

## JOIN های مختلف در SQL

قبل از اینکه با مثال ها ادامه دهیم، انواع JOIN هایی که می توانید استفاده کنید و تفاوت بین آن ها را لیست می کنیم.

JOIN: سطرها را برمی گرداند، وقتی که حداقل یک تطابق در هر دو جدول داشته باشد.

LEFT JOIN: تمام سطرهای جدول چپ را برمی گرداند؛ حتی اگر نظیر آن در جدول راست نباشد.

RIGHT JOIN: تمام سطرهای جدول راست را برمی گرداند؛ حتی اگر نظیر آن در جدول چپ نباشد.

FULL JOIN: سطرها را زمانی که نظیرش در یکی از جدول ها باشد، برمی گرداند

## کلمه کلیدی INNER JOIN در SQL

▶ کلمه کلیدی INNER JOIN اسطر ها را زمانی که حداقل یک تطابق در دو جدول وجود داشته باشد، بر می گرداند.

گرامر INNER JOIN در SQL

```
SELECT column_name(s)
FROM table_name1
INNER JOIN table_name2
ON table_name1.column_name=table_name2.column_name
```

INNER JOIN با JOIN یکسان است

مثال INNER JOIN در SQL ▶

جدول "Persons":

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

جدول "Orders":

O_Id	OrderNo	P_Id
1	77895	3
2	44678	3
3	22456	1
4	24562	1
5	34764	15

اکنون می‌فواهیم همه اشخاصی را با هر سفارشی لیست کنیم.  
از دستور SELECT زیر استفاده می‌کنیم:

```
SELECT Persons.LastName, Persons.FirstName, Orders.OrderNo
FROM Persons
INNER JOIN Orders
ON Persons.P_Id=Orders.P_Id
ORDER BY Persons.LastName
```

نتیجه اینگونه خواهد شد:

LastName	FirstName	OrderNo
Hansen	Ola	22456
Hansen	Ola	24562
Pettersen	Kari	77895
Pettersen	Kari	44678

کلمه کلیدی INNER JOIN اسطرها را زمانی که مداخل در دو جدول نظیر هستند، بر می‌گرداند. اگر سطرهایی در "Persons" هستند که نظیرشان در "Orders" نیست، آن سطرها لیست نمی‌شوند.

## کلمه کلیدی LEFT JOIN در SQL

▶ کلمه کلیدی LEFT JOIN تمام سطرهای جدول چپ را برمی گرداند (table\_name1) ؛ حتی اگر نظیرش در جدول راست (table\_name2) وجود نداشته باشد

گرامر LEFT JOIN در SQL

```
SELECT column_name(s)
FROM table_name1
LEFT JOIN table_name2
ON table_name1.column_name=table_name2.column_name
```

در بعضی از پایگاه داده ها LEFT JOIN ، LEFT OUTER JOIN نامیده می شود.

مثال LEFT JOIN در SQL

جدول "Persons":

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger



جدول "Orders":

O_Id	OrderNo	P_Id
1	77895	3
2	44678	3
3	22456	1
4	24562	1
5	34764	15

اکنون می‌فواهیم از جدول بالا همه اشخاص را با سفارشاتشان (در صورت وجود) لیست کنیم.  
از دستور SELECT زیر استفاده می‌کنیم:

```
SELECT Persons.LastName, Persons.FirstName, Orders.OrderNo
FROM Persons
LEFT JOIN Orders
ON Persons.P_Id=Orders.P_Id
ORDER BY Persons.LastName
```

نتیجه اینگونه فواید شد:

LastName	FirstName	OrderNo
Hansen	Ola	22456
Hansen	Ola	24562
Pettersen	Kari	77895
Pettersen	Kari	44678
Svendson	Tove	

کلمه کلیدی LEFT JOIN تمام سطرهای جدول چپ را برمی‌گرداند (persons)؛ حتی اگر نظیرش در جدول راست (orders) وجود نداشته باشد.

## مثال RIGHT JOIN در SQL

جدول "Persons":

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

جدول "Orders":

O_Id	OrderNo	P_Id
1	77895	3
2	44678	3
3	22456	1
4	24562	1
5	34764	15

اکنون می‌فواهیم از جدول بالا همه سفارشات با مشخصات اشخاص (در صورت وجود) را لیست کنیم.

از دستور SELECT زیر استفاده می‌کنیم:

```
SELECT Persons.LastName, Persons.FirstName, Orders.OrderNo
FROM Persons
RIGHT JOIN Orders
ON Persons.P_Id=Orders.P_Id
ORDER BY Persons.LastName
```

نتیجه اینگونه فواهد شد:

LastName	FirstName	OrderNo
Hansen	Ola	22456
Hansen	Ola	24562
Pettersen	Kari	77895
Pettersen	Kari	44678
		34764

کلمه کلیدی RIGHT JOIN تمام سطرهای جدول راست را برمی‌گرداند (Orders)؛ حتی اگر نظیرش در جدول چپ (Persons) وجود نداشته باشد.

# کلمه کلیدی FULL JOIN در SQL

کلمه کلیدی FULL JOIN سطرها را زمانی که نظیرشان در یکی از جدول ها باشد، برمی گرداند .

## گرامر FULL JOIN در SQL

```
SELECT column_name(s)
FROM table_name1
FULL JOIN table_name2
ON table_name1.column_name=table_name2.column_name
```

## مثال FULL JOIN در SQL

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

اکنون می خواهیم همه اشخاص با سفارشاتشان و همه سفارشات با اشخاص را لیست کنیم.

از دستور SELECT زیر استفاده می کنیم:

```
SELECT Persons.LastName, Persons.FirstName, Orders.OrderNo
FROM Persons
FULL JOIN Orders
ON Persons.P_Id=Orders.P_Id
ORDER BY Persons.LastName
```

نتیجه اینگونه خواهد شد:

LastName	FirstName	OrderNo
Hansen	Ola	22456
Hansen	Ola	24562
Pettersen	Kari	77895
Pettersen	Kari	44678
Svendson	Tove	
		34764

کلمه کلیدی FULL JOIN همه سطرها را از جدول پپ (persons) و همه سطرها از جدول راست (orders) بر می گرداند.

اگر سطرهایی در "Persons" وجود دارد که نظیرش در "Orders" نیست، یا اگر سطرهایی در "Orders" وجود دارد که نظیرش در "Persons" نیست، آن سطرها نیز لیست می شوند.

## محدودیت PRIMARY KEY در SQL:

محدودیت کلید اصلی مسئول شناسایی منحصر به فردی هر رکورد در جدول پایگاه داده است.

کلید اصلی باید از مقادیر منحصر به فردی برخوردار باشد.

ستون کلید اصلی نمی تواند مقدار NULL داشته باشد.

هر جدول باید یک کلید اصلی داشته باشد و هر جدول می تواند فقط یک کلید اصلی داشته باشند.

## تابع UCASE()

تابع UCASE() مقدار یک رشته را به حروف بزرگ تبدیل می کند.

## گرامر UCASE()

```
SELECT UCASE(column_name) FROM table_name
```

## گرامر برای SQL Server

```
SELECT UPPER(column_name) FROM table_name
```

## مثال UCASE() در SQL

جدول "Persons" زیر را داریم:

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

اکنون می خواهیم محتوای ستونهای "LastName" و "FirstName" بالا را انتخاب کنیم و ستون "LastName" را به حروف بزرگ تبدیل کنیم.

از دستور SQL زیر استفاده می کنیم:

```
SELECT UCASE(LastName) as LastName,FirstName FROM Persons
```

نتیجه اینگونه خواهد شد:

LastName	FirstName
HANSEN	Ola
SVENDSON	Tove
PETTERSEN	Kari

### تابع LCASE()

تابع LCASE() مقدار یک رشته را حروف کوچک تبدیل می کند.

### گرامر LCASE()

```
SELECT LCASE(column_name) FROM table_name
```

### گرامر برای SQL Server

```
SELECT LOWER(column_name) FROM table_name
```

### مثال تابع LCASE() در SQL

جدول "Persons" زیر را داریم:

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

اکنون می خواهیم محتوای "LastName" و "FirstName" ستون های بالا را انتخاب کنیم و ستون "LastName" را به حروف کوچک تبدیل کنیم.

از دستور SQL زیر استفاده می کنیم:

```
SELECT LCASE(LastName) as LastName,FirstName FROM Persons
```

نتیجه اینگونه خواهد شد:

LastName	FirstName
hansen	Ola
svendson	Tove
pettersen	Kari

Group By با بیش از یک ستون

ما می توانیم از دستور Group By برای بیش از یک ستون استفاده کنیم، مثل این:

```
SELECT Customer, OrderDate, SUM(OrderPrice) FROM Orders  
GROUP BY Customer, OrderDate
```

### عبارت Having

عبارت Having به SQL اضافه شده چون کلمه کلیدی WHERE نمی تواند برای توابع جمعی استفاده شود.

### گرامر عبارت Having

```
SELECT column_name, aggregate_function(column_name)  
FROM table_name  
WHERE column_name operator value  
GROUP BY column_name  
HAVING aggregate_function(column_name) operator value
```

### مثال Having در SQL

جدول "Orders" زیر را داریم:

O_Id	OrderDate	OrderPrice	Customer
1	2008/11/12	1000	Hansen
2	2008/10/23	1600	Nilsen
3	2008/09/02	700	Hansen
4	2008/09/03	300	Hansen
5	2008/08/30	2000	Jensen
6	2008/10/04	100	Nilsen



اکنون می خواهیم هر مشتری را که جمع سفارش کمتر از ۲۰۰۰ دارد را پیدا کنیم.

از دستور SQL زیر استفاده می کنیم:

```
SELECT Customer,SUM(OrderPrice) FROM Orders  
GROUP BY Customer  
HAVING SUM(OrderPrice)<2000
```

نتیجه اینگونه خواهد شد:

Customer	SUM(OrderPrice)
Nilsen	1700

## مثال کاراکترهای جایگزین شونده SQL

▶ جدول "Persons" زیر را داریم:

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

▶ استفاده از کاراکتر جایگزین شونده %

اکنون می خواهیم از جدول "persons" اشخاصی را انتخاب کنیم که شهر زندگی آنها با "sa" شروع می شود.

از دستور SELECT زیر استفاده می کنیم:

```
SELECT * FROM Persons  
WHERE City LIKE 'sa%'
```

نتیجه اینگونه فواید شد: ►

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes

► سپس می‌فواهیم از جدول "persons" اشفاصی را انتفاب کنیم که شهر زندگی آنها شامل الگوی "nes" باشد.

از دستور SELECT زیر استفاده می‌کنیم:

```
SELECT * FROM Persons  
WHERE City LIKE '%nes%'
```

نتیجه اینگونه فواید شد:

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes

## مثال DELETE در SQL

▶ جدول "Persons":

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger
4	Nilsen	Johan	Bakken 2	Stavanger
5	Tjessem	Jakob	Nissestien 67	Sandnes

▶ اکنون می خواهیم شخص "Tjessem, Jakob" را از جدول "Persons" حذف کنیم.

```
DELETE FROM Persons  
WHERE LastName='Tjessem' AND FirstName='Jakob'
```

▶ جدول "Persons" اکنون شبیه این است:

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger
4	Nilsen	Johan	Bakken 2	Stavanger

## گرامر ORDER By در SQL

- ▶ مثال ORDER BY
- ▶ جدول "Persons":

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger
4	Nilsen	Tom	Vingvn 23	Stavanger

- ▶ ما می‌خواهیم همه اشخاص را در جدول بالا انتخاب کنیم؛ اما می‌خواهیم آنها را بوسیله نام خانوادگی شان مرتب کنیم.

- ▶ از جمله SELECT زیر استفاده می‌کنیم:

```
SELECT * FROM Persons  
ORDER BY LastName
```

- ▶ نتیجه اینگونه خواهد شد:

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
4	Nilsen	Tom	Vingvn 23	Stavanger
3	Pettersen	Kari	Storgt 20	Stavanger
2	Svendson	Tove	Borgvn 23	Sandnes