به نام خدا

درس: مباحث ویژه

فصل اول
دستورات پیشرفته

عملگر LIKE در SQL

- عملگر LIKE برای جستجوی یک الگوی تعیین شده در یک ستون استفاده می شود .

```
SELECT column_name(s)
FROM table_name
WHERE column_name LIKE pattern
```

- مثال عملگر LIKE

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

عملگر LIKEدر SQL (ادامه)

- اکنون می خواهیم از جدول اسلاید قبل اشخاصی را که شهر زندگی آن ها با " s"شروع می شود را انتخاب کنیم.

از دستور SELECTزیر استفاده می کنیم:

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

- علامت "%" می تواند برای تعریف کاراکترهای جایگزین شونده (حروف ناپیدا در الگو) قبل و بعد از الگو استفاده شود .

عملگر LIKEدر SQL (ادامه)

نتیجه اینگونه خواهد شد

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

- سپس می خواهیم از جدول "persons" اشخاصی را انتخاب کنیم که شهر زندگی آنها با " s"پایان می یابد.

از دستور SELECTزیر استفاده می کنیم:

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

عملگر LIKEدر SQL (ادامه)

نتیجه اینگونه خواهد شد:

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

- سپس می خواهیم از جدول "persons" اشخاصی را انتخاب کنیم که شهر زندگی آنها شامل الگوی " "tav" باشد.

از دستور SELECTزیر استفاده می کنیم.

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

- نتیجه اینگونه خواهد شد:

P_Id	LastName	FirstName	Address	City
3	Pettersen	Kari	Storgt 20	Stavanger

عملگر LIKEدر SQL (ادامه)

- همچنین با استفاده از کلمه کلیدی NOTمی توانیم از جدول "Persons"اشخاصی که شهر زندگی آنها شامل الگوی "tav" نیست را انتخاب کنیم.
از دستور SELECT زیر استفاده می کنیم :

```
SELECT * FROM Persons
WHERE City NOT LIKE '%tav%'
```

- نتیجه اینگونه خواهد شد

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

کاراکترهای جایگزین شونده SQL

- کاراکترهای جایگزین شونده SQL، هنگام جستجوی داده در یک پایگاه داده می تواند یک یا چند کاراکتر را جایگزین کند.

کاراکترهای جایگزین شونده SQL باید همراه با عملگر LIKE استفاده شوند
از کاراکترهای جایگزین شونده زیر می توان در SQL استفاده کرد:

کاراکترهای جایگزین شونده	تعریف
%	یک جانشین برای صفر یا کاراکترهای بیشتر
_	یک جانشین برای دقیقا یک کاراکتر
[charlist]	هر کاراکتر تنها در Charlist (لیست کاراکترها)
[^charlist] یا [!charlist]	هرکاراکتر تنها که در Charlist نباشد

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 SELECTزیر استفاده می کنیم:

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

- نتیجه اینگونه خواهد شد:

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

- سپس می خواهیم از جدول "persons " اشخاصی را انتخاب کنیم که شهر زندگی آنها شامل الگوی "nes " باشد .

از دستور SELECTزیر استفاده می کنیم:

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

نتیجه اینگونه خواهد شد:

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

استفاده از کاراکتر جایگزین شونده _

- اکنون می خواهیم از جدول "Persons " اشخاصی که نام آنها با هر کاراکتری شروع می شود و با "la "ادامه می یابد را انتخاب کنیم.

از دستور SELECT زیر استفاده می کنیم:

```
SELECT * FROM Persons
WHERE FirstName LIKE '_la'
```

نتیجه اینگونه خواهد شد:

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes

- سپس می خواهیم از جدول " "personsاشخاصی را انتخاب کنیم که نام خانوادگی آن ها با " "sشروع می شود و با هر کاراکتر، "end"، هر کاراکتر و "on"ادامه می یابد را انتخاب کنیم.
از دستور SELECT زیر استفاده می کنیم :

```
SELECT * FROM Persons
WHERE LastName LIKE 'S_end_on'
```

نتیجه اینگونه خواهد شد:

P_Id	LastName	FirstName	Address	City
2	Svendson	Tove	Borgvn 23	Sandnes

استفاده از کاراکتر جایگزین شونده[charlist]

- جدول "persons"اشخاصی را انتخاب کنیم که نام خانوادگی آن ها با "s" یا "b”یا "p" شروع می شود .

از دستور SELECT زیر استفاده می کنیم:

```
SELECT * FROM Persons
WHERE LastName LIKE '[bsp]%'
```

نتیجه اینگونه خواهد شد:

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes

عملگر INدر 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

- اکنون می خواهیم از جدول بالا اشخاصی را که نام خانوادگی آن ها برابر با "Hansen" یا "pettersen" است را انتخاب کنیم .

از دستور SELECT زیر استفاده می کنیم:

```
SELECT * FROM Persons
WHERE LastName IN ('Hansen','Pettersen')
```

نتیجه اینگونه خواهد شد:

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

عملگر BETWEEN در SQL

- عملگر BETWEEN دامنه ای از داده ها را از بین دو مقدار انتخاب می کند. مقادیر می توانند اعداد، متن یا داده باشد.

گرامر BETWEEN در SQL

```
SELECT column_name(s)
FROM table_name
WHERE column_name
BETWEEN value1 AND value2
```

- مثال عملگر BETWEEN

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

- اکنون می خواهیم از جدول بالا اشخاصی را که نام خانوادگی آن ها به صورت حروف الفبا بین "Hansen"و "pettersen" است را انتخاب کنیم.
از دستور SELECT زیر استفاده می کنیم

```
SELECT * FROM Persons
WHERE LastName
BETWEEN 'Hansen' AND 'Pettersen'
```

نتیجه اینگونه خواهد شد:

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes

تذکر: عملگر BETWEENدر پایگاه داده های مختلف به طور متفاوت عمل می کند.

- برای نمایش اشخاصی که بیرون از دامنه مثال قبل هستند، از NOT BETWEEN استفاده کنید:

```
SELECT * FROM Persons
WHERE LastName
NOT BETWEEN 'Hansen' AND 'Pettersen'
```

نتیجه اینگونه خواهد شد:

P_Id	LastName	FirstName	Address	City
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 برخی 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 برخی 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 مقدار منحصر برای هر سطر است. هر مقدار کلید اصلی باید در داخل جدول، منحصر به فرد باشد. هدف چسباندن داده ها به یکدیگر، در سراسر جدول ها، جلوگیری ازتکرار 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؛ حتی 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 کلیدیINNER JOIN در SQL

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

گرامرINNER JOIN در SQL

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

INNERJOIN 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 می خواهیم همه اشخاصی را با هر سفارشی لیست کنیم.

از دستور SELECTزیر استفاده می کنیم :

```
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 JOINسطر ها را زمانی که حداقل در دو جدول نظیر هستند، بر می گرداند. اگر سطرهایی در "Persons" هستند که نظیرشان در "Orders" نیست، آن سطرها لیست نمی شوند.

کلمه کلیدی LEFT JOINدر 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

گرامرLEFT 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 می خواهیم از جدول بالا همه اشخاص را با سفارشاتشان (در صورت وجود) لیست کنیم.
از دستور SELECTزیر استفاده می کنیم:

```
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 خواهد 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 برمی گرداند (persons)؛ حتی اگر نظیرش درجدول راست (orders) وجود نداشته باشد.

کلمه کلیدی RIGHT JOIN در SQL

کلمه کلیدی RIGHT JOIN تمام سطرهای جدول راست را برمی گرداند(table_name2)
حتی اگر نظیرش درجدول چپ (table_name1) وجود نداشته باشد.

گرامر RIGHT JOIN در SQL

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

در بعضی از پایگاه داده هاRIGHT JOIN ، RIGHT OUTER JOIN نامیده می شود.

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

از دستور SELECT زیر استفاده می کنیم:

```
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 خواهد شد:

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

کلمه کلیدی RIGHT JOINتمام سطرهای جدول راست را برمی گرداند (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 SELECTزیر 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 چپ (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.

عملگر UNION در SQL

عملگر UNION در SQL دو یا چند دستور SELECT را ترکیب می کند.

عملگر UNION برای ترکیب جدول نتیجه، از دو یا چند عبارت SELECT استفاده می شود.

توجه کنید که هر عبارت SELECT در داخل UNION باید تعداد یکسانی از ستون ها را داشته باشد. همچنین ستون ها باید همانند انواع داده های مشابه داشته باشند. همچنین ستون ها در هر عبارت مشابه باید ترتیب یکسان داشته باشند.

توابع SQL

SQL توابع داخلی زیادی برای انجام محاسبات بر روی داده دارد.

توابع جمعی SQL

توابع جمعی SQL یک مقدار تکی محاسبه شده از مقادیر یک ستون را برمی گردانند.

توابع جمعی مفید:

- AVG() – معدل را برمی گرداند
- COUNT() – تعداد سطرها را برمی گرداند
- FIRST() – اولین مقدار را برمی گرداند
- LAST() – آخرین مقدار را برمی گرداند
- MAX() – بیشترین مقدار را برمی گرداند
- MIN() – کمترین مقدار را برمی گرداند
- SUM() – مجموع را برمی گرداند

توابع اسکالر SQL

توابع اسکالر SQL یک مقدار تکی مبنی بر مقدار ورودی را برمی گردانند.

توابع اسکالر مفید:

- UCASE() - تبدیل رشته به حروف بزرگ
- LCASE() - تبدیل رشته به حروف کوچک
- MID() - استخراج کاراکترها از یک رشته متنی
- LEN() - طول یک رشته متنی را برمی گرداند
- ROUND() - یک رشته عددی را با تعداد اعشار مشخص گرد می کند
- NOW() - تاریخ و ساعت فعلی سیستم را برمی گرداند
- FORMAT() - چگونگی نمایش یک فیلد را مشخص می کند

دستور Group By در SQL

توابع جمعی اغلب به یک دستور Group By اضافه شده نیاز دارند.

دستور Group By

دستور Group By با توابع جمعی، برای دسته بندی کردن نتیجه توسط یک یا چند ستون استفاده می شود.

گرامر Group By در SQL

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

مثال Group By 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 مجموع سفارشات هر مشتری را پیدا کنیم.

ما مجبور خواهیم شد از دستور Group By برای دسته بندی کردن مشتریان استفاده کنیم.

از دستور SQL زیر استفاده می کنیم:

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

نتیجه اینگونه خواهد شد:

Customer	SUM(OrderPrice)
Hansen	2000
Nilsen	1700
Jensen	2000

بگذارید ببینیم؛ اگر از دستور Group By صرف نظر می کردیم چه اتفاقی می افتاد:

```
SELECT Customer,SUM(OrderPrice) FROM Orders
```

نتیجه اینگونه خواهد شد:

Customer	SUM(OrderPrice)
Hansen	5700
Nilsen	5700
Hansen	5700
Hansen	5700
Jensen	5700
Nilsen	5700

نتیجه بالا چیزی نیست که می خواستیم.

تشریح اینکه چرا دستور SELECT بالا نمی تواند استفاده شود: دستور SELECT بالا دو ستون مشخص شده دارد (Customer و SUM(OrderPrice)). هنگامیکه "Customer" ۶ مقدار برمی گرداند (یک مقدار برای هر در سطر جدول "Orders")، "SUM(OrderPrice)" یک مقدار واحد برمی گرداند (آن مجموع ستون "OrderPrice" است). بنابراین این نتیجه درست را به ما نخواهد داد. با این حال، دیدید که دستور Group By این مشکل را حل می کند.

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 شد؛ 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

اکنون می خواهیم مشتریانی با نام "Hansen" یا "Jensen" که جمع سفارش بیشتر از ۱۵۰۰ دارند را پیدا کنیم.

یک عبارت WHERE معمولی به دستور SQL اضافه می کنیم:

```
SELECT Customer,SUM(OrderPrice) FROM Orders
WHERE Customer='Hansen' OR Customer='Jensen'
GROUP BY Customer
HAVING SUM(OrderPrice)>1500
```

نتیجه اینگونه خواهد شد:

Customer	SUM(OrderPrice)
Hansen	2000
Jensen	2000

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 حرو.ف 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

تابع MID()

تابع MID() برای استخراج کاراکترها از یک رشته متنی استفاده می شود.

گرامر تابع MID()

```
SELECT MID(column_name,start[,length]) FROM table_name
```

پارامتر	توضیحات
column_name	لازم. رشته مورد نظر برای استخراج کاراکترها
start	لازم. موقعیت شروع را مشخص کنید (شروع از ۱)
length	اختیاری. تعداد کاراکترها برای بازگشت. اگر صرفنظر شود، تابع MID() بقیه متن را برمی گرداند.

مثال MID() 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 چهار کاراکتر اول از ستون "City" بالا را استخراج کنیم.

از دستور SQL زیر استفاده می کنیم:

```
SELECT MID(City,1,4) as SmallCity FROM Persons
```

نتیجه اینگونه خواهد شد:

SmallCity
Sand
Sand
Stav

تابع LEN()

تابع LEN() طول مقدار یک فیلد متنی را برمی گرداند.

گرامر تابع LEN()

```
SELECT LEN(column_name) FROM table_name
```

مثال تابع LEN() 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 طول مقادیر در ستون "Address" بالا را انتخاب کنیم.

از دستور SQL زیر استفاده می کنیم:

```
SELECT LEN(Address) as LengthOfAddress FROM Persons
```

نتیجه اینگونه خواهد شد:

LengthOfAddress
12
9
9

تابع ROUDN()

تابع ROUDN() برای گرد کردن یک فیلد عددی به تعداد عدد اعشاری مشخص.

گرامر ROUDN() در SQL

```
SELECT ROUND(column_name,decimals) FROM table_name
```

پارامترها	توضیحات
column_name	لازم. فیلدی که باید گرد شود.
decimals	لازم. تعداد اعشار برای گرد شدن را تعیین می کند.

مثال ROUDN() در SQL

جدول "Products" زیر را داریم:

Prod_Id	ProductName	Unit	UnitPrice
1	Jarlsberg	1000 g	10.45
2	Mascarpone	1000 g	32.56
3	Gorgonzola	1000 g	15.67

اکنون می خواهیم نام محصولات و قیمت گرد شده به نزدیکترین عدد نمایش دهیم.

از دستور SQL زیر استفاده می کنیم:

```
SELECT ProductName, ROUND(UnitPrice,0) as UnitPrice FROM Products
```

نتیجه اینگونه خواهد شد:

ProductName	UnitPrice
Jarlsberg	10
Mascarpone	33
Gorgonzola	16

تابع NOW()

تابع NOW() تاریخ و زمان فعلی سیستم را برمی گرداند.

گرامر NOW() در SQL

```
SELECT NOW() FROM table_name
```

مثال NOW()

جدول "Products" زیر را داریم:

Prod_Id	ProductName	Unit	UnitPrice
1	Jarlsberg	1000 g	10.45
2	Mascarpone	1000 g	32.56
3	Gorgonzola	1000 g	15.67

اکنون می خواهیم محصولات و قیمت ها در تاریخ امروز را نمایش دهیم.

از دستور SQL زیر استفاده می کنیم:

```
SELECT ProductName, UnitPrice, Now() as PerDate FROM Products
```

نتیجه اینگونه خواهد شد:

ProductName	UnitPrice	PerDate
Jarlsberg	10.45	10/7/2008 11:25:02 AM
Mascarpone	32.56	10/7/2008 11:25:02 AM
Gorgonzola	15.67	10/7/2008 11:25:02 AM

تابع FORMAT()

تابع FORMAT() برای چگونگی نمایش فرمت یک فیلد استفاده می شود.

گرامر FORMAT() در SQL

```
SELECT FORMAT(column_name,format) FROM table_name
```

پارامتر	توضیحات
column_name	لازم. فیلدی که باید فرمت دهی شود.
format	لازم. فرمت را تعیین می کند.

مثال FORMAT() در SQL

جدول "Products" زیر را داریم:

Prod_Id	ProductName	Unit	UnitPrice
1	Jarlsberg	1000 g	10.45
2	Mascarpone	1000 g	32.56
3	Gorgonzola	1000 g	15.67

اکنون می خواهیم محصولات و قیمت در تاریخ امروز (با فرمت "YYYY-MM-DD") را نمایش دهیم.

از دستور SQL زیر استفاده می کنیم:

```
SELECT ProductName, UnitPrice, FORMAT(Now(),'YYYY-MM-DD') as PerDate FROM Products
```

نتیجه اینگونه خواهد شد:

ProductName	UnitPrice	PerDate
Jarlsberg	10.45	2008-10-07
Mascarpone	32.56	2008-10-07
Gorgonzola	15.67	2008-10-07