

SELECT

OrderH.invoiceNo, OrderH.invoiceData, OrderH.customerCode, OrderD.itemCode, I.itemName, OrderD.qty, OrderD.netPrice, OrderD

FROM

OrderHeader AS OrderH

INNER JOIN Customer AS Cust ON OrderH.customerCode = Cust.customerCode

INNER JOIN OrderDetail AS OrderD ON OrderH.orderCode = OrderD.orderCode

INNER JOIN Item AS I ON OrderD.itemCode = I.itemCode

WHERE

OrderD.netPrice > 1000

«پیش نمایش»

کتاب آموزش SQL

همراه با صدها مثال

- ◀ SQL سرنام واژگان Structured Query Language به معنی زبان جستجوی ساخت یافته است.
- SQL به شما اجازه دسترسی و دستکاری پایگاه داده را می دهد.
- ◀ SQL را می توان به دو بخش تقسیم کرد: زبان دستکاری داده Data Manipulation Language یا DML و زبان تعریف داده Data Definition Language یا DDL
- ◀ توجه داشته باشید که مطالب گردآوری شده این کتاب، کاملاً بصورت فارسی و همراه با صدها مثال ارائه شده است.

نسخه 1.2

فصل ۱: آموزش مقدماتی SQL

۶

۱-۱ - مقدمه آموزش SQL	۶
۱-۲ - دستور زبان SQL	۸
۱-۳ - دستور Select در SQL	۱۰
۱-۴ - عبارت Distinct در SQL	۱۲
۱-۵ - عبارت Where در SQL	در فایل اصلی
۱-۶ - عبارت AND-OR در SQL	در فایل اصلی
۱-۷ - عبارت Order By در SQL	در فایل اصلی
۱-۸ - دستور Insert در SQL	در فایل اصلی
۱-۹ - دستور Update در SQL	در فایل اصلی
۱-۱۰ - دستور Delete در SQL	در فایل اصلی

فصل ۲: آموزش پیشرفته SQL

۱۴

۲-۱ - عملگر Like در SQL	۱۴
۲-۲ - کاراکترهای جایگزین (Wildcard) چیست؟	۱۶
۲-۳ - عبارت TOP در SQL	۲۰
۲-۴ - عملگر IN در SQL	۲۳
۲-۵ - عملگر Between در SQL	در فایل اصلی
۲-۶ - نام مستعار (AS) در SQL	در فایل اصلی
۲-۷ - پیوند بین جداول (Join) در SQL	در فایل اصلی
۲-۸ - عبارت Inner Join در SQL	در فایل اصلی

۹-۲ - عبارت Left Join در SQL.....	در فایل اصلی
۱۰-۲ - عبارت Right Join در SQL.....	در فایل اصلی
۱۱-۲ - عبارت Full Join در SQL.....	در فایل اصلی
۱۲-۲ - عبارت Union در SQL.....	در فایل اصلی
۱۳-۲ - دستور Select Into در SQL.....	در فایل اصلی
۱۴-۲ - دستور Create DB در SQL.....	در فایل اصلی
۱۵-۲ - دستور Create Table در SQL.....	در فایل اصلی
۱۶-۲ - محدودیت (Constraint) چیست؟.....	در فایل اصلی
۱۷-۲ - محدودیت Not Null در SQL.....	در فایل اصلی
۱۸-۲ - محدودیت Unique در SQL.....	در فایل اصلی
۱۹-۲ - کلید اصلی (Primary Key) چیست؟.....	در فایل اصلی
۲۰-۲ - کلید خارج (Foreign Key) چیست؟.....	در فایل اصلی
۲۱-۲ - محدودیت Check در SQL.....	در فایل اصلی
۲۲-۲ - محدودیت Default در SQL.....	در فایل اصلی
۲۳-۲ - دستور Create Index در SQL.....	در فایل اصلی
۲۴-۲ - دستور Drop در SQL.....	در فایل اصلی
۲۵-۲ - دستور Alter در SQL.....	در فایل اصلی
۲۶-۲ - افزایش خودکار (Auto-Increment) در SQL.....	در فایل اصلی
۲۷-۲ - کاربرد View در SQL.....	در فایل اصلی
۲۸-۲ - تاریخ ها در SQL.....	در فایل اصلی
۲۹-۲ - عبارت IS Null در SQL.....	در فایل اصلی
۳۰-۲ - تابع ISNULL در SQL.....	در فایل اصلی
۳۱-۲ - انواع داده در SQL.....	در فایل اصلی

۲۵.....	۳-۱ - توابع در SQL
۲۶.....	۳-۲ - تابع AVG در SQL
۲۷.....	۳-۳ - تابع COUNT در SQL
۳۰.....	۳-۴ - تابع FIRST در SQL
در فایل اصلی.....	۳-۵ - تابع LAST در SQL
در فایل اصلی.....	۳-۶ - تابع MAX در SQL
در فایل اصلی.....	۳-۷ - تابع MIN در SQL
در فایل اصلی.....	۳-۸ - تابع SUM در SQL
در فایل اصلی.....	۳-۹ - عبارت Group By در SQL
در فایل اصلی.....	۳-۱۰ - عبارت Having در SQL
در فایل اصلی.....	۳-۱۱ - تابع UCASE در SQL
در فایل اصلی.....	۳-۱۲ - تابع LCASE در SQL
در فایل اصلی.....	۳-۱۳ - تابع MID در SQL
در فایل اصلی.....	۳-۱۴ - تابع LEN در SQL
در فایل اصلی.....	۳-۱۵ - تابع ROUND در SQL
در فایل اصلی.....	۳-۱۶ - تابع NOW در SQL
در فایل اصلی.....	۳-۱۷ - تابع FORMAT در SQL
در فایل اصلی.....	۳-۱۸ - تابع LEFT در SQL
در فایل اصلی.....	۳-۱۹ - تابع RIGHT در SQL
در فایل اصلی.....	۳-۲۰ - تابع CAST در SQL

۳۲	۴-۱ - لیست کامل دستورات در SQL
در فایل اصلی	۴-۲ - میزبانی SQL (SQL Hosting)
در فایل اصلی	۴-۳ - خلاصه مباحث SQL

فصل ۱ : آموزش مقدماتی SQL

۱-۱ - مقدمه آموزش SQL

SQL (اس کیو ال) یک زبان استاندارد برای دسترسی به پایگاه داده می باشد.

در این آموزش، چگونگی دسترسی به داده ها و دستکاری آنها را با استفاده از [SQL](#) فراخواهید گرفت.

SQL چیست؟

- SQL مخفف Structured Query Language به معنی زبان جستجوی ساخت یافته است.
- SQL به شما اجازه دسترسی و دستکاری پایگاه داده را می دهد.
- SQL یک استاندارد (ANSI American National Standards Institute) می باشد.

در این کتاب چه چیزهایی را فراخواهید گرفت؟

- در این آموزش می توانید queryهایی (پرس و جوهای) را بر روی پایگاه داده اجرا کنید.
- در این آموزش می توانید داده ها را از پایگاه داده بازیابی کنید.
- در این آموزش می توانید رکوردهایی را در پایگاه داده درج (insert) کنید.
- در این آموزش می توانید رکوردهایی را در پایگاه داده تغییر دهید (به روز رسانی کنید).
- در این آموزش می توانید رکوردهایی را از پایگاه داده حذف (delete) کنید.
- در این آموزش می توانید پایگاه داده جدیدی را ایجاد کنید.
- در این آموزش می توانید جدول جدیدی را در پایگاه داده ایجاد کنید.
- در این آموزش می توانید stored procedureها (رویه های ذخیره شده) را در پایگاه داده ایجاد کنید.
- در این آموزش می توانید viewها را در پایگاه داده ایجاد کنید.
- در این آموزش می توانید سطح دسترسی روی جداول، procedureها و viewها را تنظیم کنید.

SQL یک استاندارد است اما...

گرچه SQL یک استاندارد ANSI می باشد و نسخه های متفاوت زیادی از آن وجود دارد، اما همه نسخه های آن از فرامین اصلی حمایت می کنند. (مثل فرامین (SELECT, UPDATE, DELETE, INSERT, WHERE) توجه: بیشتر برنامه های پایگاه داده علاوه بر استاندارد SQL الحاقات اختصاصی خود را دارند.

آموزش استفاده از SQL در وب سایت

برای ساخت یک وب سایت که داده ها را از یک پایگاه داده نشان دهد به موارد زیر نیاز دارید:

- یک برنامه سیستم مدیریت پایگاه داده رابطه ای مثل MySQL, [SQL Server](#), MS Access
- زبان های سمت سرور مثل [PHP](#) یا [ASP](#)
- SQL
- [HTML /CSS](#)

آموزش RDBMS

RDBMS مخفف Relational Database Management System به معنای سیستم مدیریت پایگاه داده رابطه ای است.

RDBMS یا سیستم مدیریت پایگاه داده رابطه ای، اساسی است برای SQL و همچنین تمام سیستم های پایگاه داده ای مدرن مثل:

Microsoft Access و MS SQL Server, IBM DB2, Oracle, **MySQL**

داده ها در RDBMS ها روی اشیایی از پایگاه داده به نام جدول ها ذخیره می شوند.

یک جدول مجموعه ای از داده های مرتبط به هم می باشد و شامل سطر و ستون ها می باشد.

۱-۲- دستور زبان SQL

جداول پایگاه داده

یک پایگاه داده از یک یا چند جدول تشکیل می شود. هر جدول با یک نام مشخص می شود. (مثلا "Customers" یا "Orders") جداول حاوی رکوردها یا سطرهای حاوی داده می باشند.

در زیر جدولی با نام "Persons" را مشاهده می کنید:

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

جدول فوق شامل سه رکورد (هر رکورد برای یک فرد) و پنج ستون (P_Id , LastName , FirstName , Address and City) می باشد.

عبارات SQL:

غالب عملیات هایی که نیاز دارید روی یک پایگاه داده انجام شود توسط عبارات SQL انجام می شوند.

عبارت SQL زیر، تمام رکوردهای جدول "Persons" را انتخاب می کند:

```
SELECT * FROM Persons
```

در این خودآموز همه چیز در مورد عبارات SQL بیاموز داده خواهد شد.

به یاد داشته باشید که ...

- SQL به کوچکی و بزرگی حروف حساس نمی باشد. (case sensitive نمی باشد)

سمیکالن ";" بعد از عبارات SQL نیاز است یا نه؟

بعضی از سیستم های پایگاه داده به سَمیکالن ";" در انتهای عبارات SQL نیاز دارند.

سمیکالن ";" یک روش استاندارد برای جدا کردن عبارات SQL در سیستمهای پایگاه داده ای می باشد که اجازه می دهند بیش از یک عبارت SQL در یک ارتباط به سرور اجرا شوند.

ما از MS Access و SQL Server استفاده می کنیم و نیازی نیست که بعد از هر عبارت SQL سَمیکالن بگذاریم اما در بعضی از برنامه های پایگاه داده ما باید از سَمیکالن استفاده کنیم.

SQL DDL و SQL DML

SQL را می توان به دو بخش تقسیم کرد: زبان دستکاری داده Data Manipulation Language یا DML و زبان تعریف داده Data Definition Language یا DDL.

بخش SQL DML:

- SELECT: داده را از یک پایگاه داده بیرون می کشد.
- UPDATE: داده ای را در یک پایگاه داده به روز رسانی می کند(تغییر می دهد).
- DELETE: داده را از یک پایگاه داده حذف می کند.
- INSERT INTO: داده جدیدی را به پایگاه داده اضافه می کند.

بخش SQL DDL: این بخش از دستورات SQL اجازه می دهد تا جداول پایگاه داده ایجاد یا حذف شوند. همچنین شاخصها (indexes) و کلیدها را تعریف می کند، ارتباط بین جداول و محدودیت ها را مشخص می کند.

- CREATE DATABASE: یک پایگاه داده جدید ایجاد می کند.

- ALTER DATABASE: یک پایگاه داده را تغییر می دهد.
- CREATE TABLE: یک جدول جدید ایجاد می کند.
- ALTER TABLE: یک جدول را تغییر می دهد.
- DROP TABLE: یک جدول را حذف می کند.
- CREATE INDEX: یک شاخص یا index (کلید جستجو) ایجاد می کند.
- DROP INDEX: یک شاخص را حذف می کند.

۱-۳- دستور Select در SQL

دستور SELECT برای انتخاب و بیرون کشیدن داده از پایگاه داده استفاده می شود.

نتایج در جدول نتایج که result-set نامیده می شود ذخیره می شود.

فرم دستور SELECT:

```
SELECT column_name(s)
FROM table_name
```

```
SELECT * FROM table_name
```

💡 توجه: SQL حساس به کوچکی یا بزرگی حروف نمی باشد. SELECT دقیقاً معادل select است.

مثال: در جدول Persons

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

می خواهیم محتوای ستون های "LastName" و "FirstName" را از جدول بالا انتخاب کنیم.

از دستور SELECT زیر استفاده می کنیم:

```
SELECT LastName,FirstName FROM Persons
```

FirstName	LastName
Ola	Hansen
Tove	Svendson
Kari	Pettersen

مثال * Select

می خواهیم تمام ستون های جدول Persons را انتخاب کنیم.

از دستور زیر استفاده می کنیم:

```
SELECT LastName,FirstName FROM Persons
```

result-set به صورت زیر نمایش داده خواهد شد:

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

پیمایش در result-set یا جدول نتایج:

اکثر سیستم های نرم افزاری پایگاه داده اجازه پیمایش در رکوردهای جداول نتیجه را به وسیله توابع می دهند
مثل:

Move-To-First-Record, Get-Record-Content, Move-To-Next-Record

این توابع جز این مجموعه بیاموزی نمی باشد. برای یادگیری این توابع می توانید لینک [PHP Home](#) را بررسی کنید.

۱-۴- عبارت Distinct در SQL

در این فصل دستور SELECT DISTINCT را توضیح می دهیم.

دستور SELECT DISTINCT

در یک جدول بعضی ستونها ممکن است داده های تکراری داشته باشند. بعضی مواقع شما می خواهید لیستی تهیه کنید که تنها داده های یکتا در آن باشد. یعنی همه داده ها را نشان دهد و داده های تکراری را یکبار نشان دهد.

کلید واژه DISTINCT برای برگرداندن داده های یکتا و متمایز به کار برده می شود.

فرم دستور SELECT DISTINCT:

```
SELECT DISTINCT column_name(s)
FROM table_name
```

مثال:

در جدول زیر می خواهیم داده های یکتا از ستون City را بدست آوریم:

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

از عبارت زیر برای این منظور استفاده می کنیم:

پایان پیش نمایش این فصل، ادامه در فایل اصلی ...!

توجه: پیش نمایش بالا، تنها با هدف جلب اطمینان شما از محتوای اصلی تهیه شده است. بنابراین اگر مورد تأیید شما قرار گرفت لطفاً از طریق سایت Beyamooz.com اقدام به خرید فایل کامل آن نمایید.

لینک خرید کتاب های آموزشی

۱. [کتاب آموزش SQL](#) (کتاب پیش رو)

۲. [کتاب آموزش Bootstrap](#)

۳. [کتاب آموزش JavaScript](#)

۴. [کتاب آموزش HTML](#)

۵. [کتاب آموزش CSS](#)

موفق و پاینده باشید

فصل ۲: آموزش پیشرفته SQL

۲-۱ - عملگر Like در SQL

عملگر LIKE در بند WHERE برای پیدا کردن یک الگوی خاص در یک ستون استفاده می شود.

فرم عملگر LIKE:

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

مثال عملگر LIKE

در جدول "Persons"

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

می خواهیم افرادی که در شهری زندگی می کنند که با حرف S شروع می شود را انتخاب کنیم.

از دستور زیر استفاده می کنیم:

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

توجه کنید که یک الگو را در بین کوتیشن می نویسیم.

علامت %/ جانشینی برای یک یا چند کاراکتر در الگو استفاده می شود که می تواند این کاراکترها هر چیزی باشد.

جدول نتایج بدین شکل خواهد بود:

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

حال اگر بخواهیم افرادی را انتخاب کنیم که در شهری زندگی می کنند که به کاراکتر S ختم می شوند از دستور زیر استفاده می کنیم:

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

جدول نتایج به شکل زیر خواهد بود:

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

حال اگر بخواهیم افرادی را انتخاب کنیم که در شهری زندگی می کنند که حاوی کلمه "tav" در نام شهر خود می باشند از دستور زیر استفاده می کنیم:

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

جدول نتایج به شکل زیر خواهد بود:

City	Address	FirstName	LastName	P_Id
Stavanger	Storgt 20	Kari	Pettersen	3

می شود افرادی را انتخاب کرد که در شهری زندگی می کنند که حاوی کلمه "tav" نمی باشند. این کار را با کلید واژه NOT قبل از عملگر LIKE انجام می دهیم. (NOT LIKE).

از دستور زیر استفاده می کنیم:

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

جدول نتایج به شکل زیر خواهد بود:

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

۲-۲- کاراکترهای جایگزین (Wildcard) چیست؟

wildcard ها می توانند هنگام جستجوی داده از یک پایگاه داده مورد استفاده قرار گیرند.

Wildcard ها یا کاراکترهای جایگزین در SQL

Wildcard ها می توانند جانشین یک یا چند کاراکتر برای جستجوی داده در یک پایگاه داده شوند.

Wildcard ها باید به همراه عملگر LIKE استفاده شوند.

Wildcardهای زیر در SQL می توانند استفاده شوند:

توضیحات	Wildcard
جانشینی برای صفر یا چند کاراکتر	%
جانشینی برای دقیقا یک کاراکتر	_
کاراکترهای نوشته شده در براکت	[charlist]
به غیر از کاراکترهای نوشته شده در براکت	[^charlist] یا [!charlist]

مثال Wildcardها

در جدول Persons

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

کاربرد %

می خواهیم افرادی را انتخاب کنیم که در شهری زندگی می کنند که با sa شروع می شود.

از دستور زیر استفاده می کنیم:

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

جدول نتایج به شکل زیر خواهد بود:

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

حال می خواهیم افرادی را انتخاب کنیم که در شهری زندگی می کنند که نام آن شهر حاوی nes می باشد.
از دستور زیر استفاده می کنیم:

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

جدول نتایج به شکل زیر خواهد بود:

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

کاربرد _ (زیر خط)

می خواهیم افرادی را انتخاب کنیم که نام کوچک (first name) آنها با هر کاراکتری شروع شود و در ادامه حروف la آمده باشد.

از دستور زیر استفاده می کنیم:

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

جدول نتایج به شکل زیر خواهد بود:

City	Address	FirstName	LastName	P_Id
Sandnes	Timoteivn 10	Ola	Hansen	1

حال می خواهیم افرادی را انتخاب کنیم که نام خانوادگی (last name) آنها با s شروع می شود و بعد از آن یک کاراکتر باشد و در ادامه end باشد و بعد یک کاراکتر باشد و در ادامه on باشد.

از دستور زیر استفاده می کنیم:

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

جدول نتایج به شکل زیر خواهد شد:

City	Address	FirstName	LastName	P_Id
Sandnes	Borgvn 23	Tove	Svendson	2

کاربرد [آرایه ای از کاراکترها]

می خواهیم از جدول persons افرادی را انتخاب کنیم که نام خانوادگی آنها با "b" یا "s" یا "p" شروع می شود.

از دستور زیر استفاده می کنیم:

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

جدول نتایج به شکل زیر خواهد بود:

City	Address	FirstName	LastName	P_Id
Sandnes	Borgvn 23	Tove	Svendson	2
Stavanger	Storgt 20	Kari	Pettersen	3

می خواهیم از جدول persons افرادی را انتخاب کنیم که نام خانوادگی آنها با "b" یا "s" یا "p" شروع نشود.

از دستور زیر استفاده می کنیم:

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

جدول نتایج به شکل زیر خواهد شد:

City	Address	FirstName	LastName	P_Id
Sandnes	Timoteivn 10	Ola	Hansen	1

۲-۳- عبارت TOP در SQL

عبارت TOP برای مشخص کردن تعداد رکوردهایی که می خواهیم برگردانده شود استفاده می شود.

عبارت TOP در جداول بزرگ با هزاران رکورد می تواند بسیار مفید باشد. بیرون کشیدن تعداد زیادی از رکوردها می تواند روی عملکرد بسیار مؤثر باشد.

💡 نکته: تمام سیستم های پایگاه داده از TOP پشتیبانی نمی کنند.

فرم عبارت TOP:

```
SELECT TOP number|percent column_name(s)
FROM table_name
```

معادل SELECT TOP در MySQL و Oracle

فرم آن در MySQL:

```
SELECT TOP number|percent column_name(s)
FROM table_name
```

مثال:

```
SELECT *
FROM Persons
LIMIT 5
```

فرم آن در Oracle:

```
SELECT column_name(s)
FROM table_name
WHERE ROWNUM <= number
```

مثال:

```
SELECT *  
FROM Persons  
WHERE ROWNUM <=5
```

مثال در SQL

در جدول Persons

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

می خواهیم تنها دو رکورد اول را از جدول بالا انتخاب کنیم.

از دستور زیر استفاده می کنیم:

```
SELECT TOP 2 * FROM Persons
```

جدول نتایج result-set بدین صورت خواهد شد:

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

مثال TOP PERCENT

در جدول Persons:

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

می خواهیم در جدول بالا ۵۰ درصد رکوردها را انتخاب کنیم.

از دستور زیر استفاده می کنیم:

```
SELECT TOP 50 PERCENT * FROM Persons
```

جدول نتایج به شکل زیر خواهد شد:

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

۲-۴- عملگر IN در SQL

عملگر IN به شما این امکان را می دهد که چندین ارزش را در عبارت WHERE مشخص کنید.

💡 نکته اضافه: از NOT IN هم می توانید استفاده کنید.

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

مثال:

در جدول: Persons

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

می خواهیم افرادی را که نام خانوادگی آنها "Hansen" یا "Pettersen" می باشد را انتخاب کنیم:

پایان پیش نمایش این فصل، ادامه در فایل اصلی ...!

توجه: پیش نمایش بالا، تنها با هدف جلب اطمینان شما از محتوای اصلی تهیه شده است. بنابراین اگر مورد تأیید شما قرار گرفت لطفاً از طریق سایت Beyamooz.com اقدام به خرید فایل کامل آن نمایید.

لینک خرید کتاب های آموزشی

۱. [کتاب آموزش SQL](#) (کتاب پیش رو)

۲. [کتاب آموزش Bootstrap](#)

۳. [کتاب آموزش CSS](#)

موفق و پاینده باشید

فصل ۳: آموزش توابع SQL

۳-۱ - توابع در SQL

SQL توابع تعریف شده زیادی برای انجام محاسبات بر روی داده ها دارد.

توابع Aggregate:

توابع Aggregate تنها یک مقدار را براساس داده های یک ستون برمی گردانند.

توابع پرکاربرد Aggregate:

- AVG - میانگین مقادیر یک ستون را بر می گرداند.
 - COUNT - تعداد سطرهای یک ستون را برمی گرداند.
 - FIRST - اولین مقدار ستون را بر می گرداند.
 - LAST - آخرین مقدار ستون را بر می گرداند.
 - MAX - بزرگترین مقدار ستون را بر می گرداند.
 - MIN - کوچکترین مقدار ستون را بر می گرداند.
 - SUM - مجموع داده های ستون را بر می گرداند.
-

توابع Scalar:

توابع Scalar تنها یک مقدار را بر اساس داده ورودی برمی گردانند.

توابع پرکاربرد Scalar:

- UCASE: یک فیلد را به حروف بزرگ (upper case) تبدیل می کند.
- LCASE - یک فیلد را به حروف کوچک (lower case) تبدیل می کند.
- MID - کاراکترهایی را از یک فیلد متنی بیرون می کشد.

- - LEN طول یک فیلد متنی را بر می گرداند.
- - ROUND یک فیلد عددی را به عددی با تعداد اعشار مشخص گرد می کند.
- - NOW تاریخ و ساعت جاری سیستم را بر می گرداند.
- - FORMAT مشخص می کند یک فیلد با چه فرمتی نمایش داده شود.
- - LEFT این تابع از سمت چپ یک عبارت، به تعدادی مشخص، کاراکتر برمی گرداند.
- - RIGHT این تابع از سمت راست یک عبارت، به تعدادی مشخص، کاراکتر برمی گرداند.
- - CONVERT & CAST یک عبارت را به نوعی دیگر تبدیل می کند.

توابع Aggregate و Scalar به صورت کامل در فصل های بعد توضیح داده خواهند شد.

۳-۲- تابع AVG در SQL

تابع Avg میانگین مقادیر یک ستون عددی را بر می گرداند.

نحوه استفاده:

```
SELECT AVG(column_name) FROM table_name
```

مثال:

جدول Orders را در زیر داریم:

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

می خواهیم میانگین فیلد OrderPrice را به دست آوریم.

از دستور زیر استفاده می کنیم:

```
SELECT AVG(OrderPrice) AS OrderAverage FROM Orders
```

OrderAverage
950

می خواهیم مشتریانی را پیدا کنیم که مقدار سفارش آنها بزرگتر از میانگین باشد.

از دستور زیر استفاده می کنیم:

```
SELECT Customer FROM Orders  
WHERE OrderPrice > (SELECT AVG(OrderPrice) FROM Orders)
```

نتیجه به شکل زیر خواهد بود:

Customer
Hansen
Nilsen
Jensen

۳-۳- تابع COUNT در SQL

تابع COUNT تعداد سطرهایی را بر می گرداند که در شرط صدق می کنند.

فرم تابع COUNT(column_name)

تابع COUNT(column_name) تعداد مقادیر ستون مشخص شده را بر می گرداند (مقدار NULL شمرده نمی شود).

```
SELECT COUNT(column_name) FROM table_name
```

تابع COUNT(*) تعداد رکوردهای یک جدول را بر می گرداند.

```
SELECT COUNT(*) FROM table_name
```

تابع COUNT (DISTINCT column_name) تعداد مقادیر متمایز (داده های تکراری را یک بار حساب می کند) از یک ستون را بر می گرداند.

```
SELECT COUNT(DISTINCT column_name) FROM table_name
```

تابع COUNT(DISTINCT) در ORACLE و Microsoft SQL Server کار می کند و در Microsoft Access کار نمی کند.

مثالی از COUNT(column_name)

جدول Orders را در زیر داریم:

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

می خواهیم تعداد سفارشات مشتری با نام Nilsen را بیابیم:

از دستور زیر استفاده می کنیم:

```
SELECT COUNT(Customer) AS CustomerNilsen FROM Orders  
WHERE Customer='Nilsen'
```

CustomerNilsen
2

مثالی از COUNT(*)

اگر عبارت WHERE را از دستور بالا حذف کنیم، به این صورت:

```
SELECT COUNT(*) AS NumberOfOrders FROM Orders
```

نتیجه چنین خواهد بود:

NumberOfOrders
6

که تعداد کل سطرهای جدول را بر می گرداند.

مثالی از COUNT(DISTINCT column_name)

می خواهیم تعداد مشتریان را در جدول Orders به دست آوریم (داده های تکراری یک بار شمرده شوند).
از دستور زیر استفاده می کنیم:

```
SELECT COUNT(DISTINCT Customer) AS NumberOfCustomers FROM
Orders
```

NumberOfCustomers
3

که تعداد مشتریان منحصر به فرد در جدول Orders می باشد. (Hansen, Nilsen, Jensen)

۳-۴- تابع FIRST در SQL

تابع FIRST اولین داده را از ستون انتخابی بر می گرداند.

فرم دستور FIRST:

```
SELECT FIRST(column_name) FROM table_name
```

مثال:

جدول Orders را در زیر داریم:

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

می خواهیم اولین مقدار از ستون OrderPrice را پیدا کنیم:

پایان پیش نمایش این فصل، ادامه در فایل اصلی ...!

توجه: پیش نمایش بالا، تنها با هدف جلب اطمینان شما از محتوای اصلی تهیه شده است. بنابراین اگر مورد تأیید شما قرار گرفت لطفاً از طریق سایت Beyamooz.com اقدام به خرید فایل کامل آن نمایید.

لینک خرید کتاب های آموزشی

۱. [کتاب آموزش SQL](#) (کتاب پیش رو)

۲. [کتاب آموزش Bootstrap](#)

۳. [کتاب آموزش JavaScript](#)

۴. [کتاب آموزش HTML](#)

۵. [کتاب آموزش CSS](#)

موفق و پاینده باشید

فصل ۴: خلاصه مطالب SQL

۴-۱ - لیست کامل دستورات در SQL

نحوه استفاده	دستورات SQL
SELECT column_name(s) FROM table_name WHERE condition AND OR condition	AND / OR
ALTER TABLE table_name ADD column_name datatype or ALTER TABLE table_name DROP COLUMN column_name	ALTER TABLE
SELECT column_name AS column_alias FROM table_name or SELECT column_name FROM table_name AS table_alias	AS (alias)
SELECT column_name(s) FROM table_name WHERE column_name BETWEEN value1 AND value2	BETWEEN
CREATE DATABASE database_name	CREATE DATABASE
CREATE TABLE table_name (column_name1 data_type, column_name2 data_type, column_name2 data_type, ...)	CREATE TABLE
CREATE INDEX index_name ON table_name (column_name)	CREATE INDEX

or CREATE UNIQUE INDEX index_name ON table_name (column_name)	
CREATE VIEW view_name AS SELECT column_name(s) FROM table_name WHERE condition	CREATE VIEW
DELETE FROM table_name WHERE some_column=some_value or DELETE FROM table_name	DELETE
DROP DATABASE database_name	DROP DATABASE
DROP INDEX table_name.index_name (SQL Server) DROP INDEX index_name ON table_name (MS Access) DROP INDEX index_name (DB2/Oracle) ALTER TABLE table_name DROP INDEX index_name (MySQL)	DROP INDEX
DROP TABLE table_name	DROP TABLE
SELECT column_name, aggregate_function(column_name) FROM table_name WHERE column_name operator value GROUP BY column_name	GROUP BY
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	HAVING
SELECT column_name(s) FROM table_name WHERE column_name IN (value1,value2,..)	IN
INSERT INTO table_name VALUES (value1, value2, value3,...) or INSERT INTO	INSERT INTO

table_name (column1, column2, column3,...) VALUES (value1, value2, value3,...)	
SELECT column_name(s) FROM table_name1 INNER JOIN table_name2 ON table_name1.column_name=table_name2.column_name	INNER JOIN
SELECT column_name(s) FROM table_name1 LEFT JOIN table_name2 ON table_name1.column_name=table_name2.column_name	LEFT JOIN
SELECT column_name(s) FROM table_name1 RIGHT JOIN table_name2 ON table_name1.column_name=table_name2.column_name	RIGHT JOIN
SELECT column_name(s) FROM table_name1 FULL JOIN table_name2 ON table_name1.column_name=table_name2.column_name	FULL JOIN
SELECT column_name(s) FROM table_name WHERE column_name LIKE pattern	LIKE
SELECT column_name(s) FROM table_name ORDER BY column_name [ASC DESC]	ORDER BY
SELECT column_name(s) FROM table_name	SELECT
SELECT * FROM table_name	SELECT *
SELECT DISTINCT column_name(s) FROM table_name	SELECT DISTINCT

SELECT * INTO new_table_name [IN externaldatabase] FROM old_table_name <i>or</i> SELECT column_name(s) INTO new_table_name [IN externaldatabase] FROM old_table_name	SELECT INTO
SELECT TOP number percent column_name(s) FROM table_name	SELECT TOP
TRUNCATE TABLE table_name	TRUNCATE TABLE
SELECT column_name(s) FROM table_name1 UNION SELECT column_name(s) FROM table_name2	UNION
SELECT column_name(s) FROM table_name1 UNION ALL SELECT column_name(s) FROM table_name2	UNION ALL
UPDATE table_name SET column1=value, column2=value,... WHERE some_column=some_value	UPDATE
SELECT column_name(s) FROM table_name WHERE column_name operator value	WHERE

پایان پیش نمایش این فصل، ادامه در فایل اصلی ...!

توجه: پیش نمایش بالا، تنها با هدف جلب اطمینان شما از محتوای اصلی تهیه شده است. بنابراین اگر مورد تأیید شما قرار گرفت لطفاً از طریق سایت Beyamooz.com اقدام به خرید فایل کامل آن نمایید.

لینک خرید کتاب های آموزشی

۱. [کتاب آموزش SQL](#) (کتاب پیش رو)

۲. [کتاب آموزش Bootstrap](#)

۳. [کتاب آموزش JavaScript](#)

۴. [کتاب آموزش HTML](#)

۵. [کتاب آموزش CSS](#)

موفق و پاینده باشید

همراهان خوبی باشید و با

ری پست کردن ( Repost) این محتوی در [لینکدین پیاموز](#)

ما را در تولید دوره های آموزشی مفید

حمایت کنید.

از طریق صفحه لینکدین پیاموز، ما را **فالو** کنید.

+ Follow

برای باخبر شدن از آخرین دوره های آموزشی، کافیه در

لینکدین پیاموز روی **آیکن زنگ** کلیک کنید.



Beyamooz

