Database Management System

Associate Professor Dr. Raed Ibraheem Hamed

University of Human Development,
College of Science and Technology

2015 – 2016

What is SQL?

- Structured Query Language
- Communicate with databases
- Used to created and edit databases.
- Also used to create queries, forms, and reports

Table Basics

- A Table is an object
- Database data is stored in Tables
- Each table has a unique name
- Columns have various attributes, such as column name and data type
- Rows contain records or data for the columns

Weather Sample Table

City	State	High	Low
Phoenix	Arizona	105	90
Tuscon	Arizona	101	92
Flagstaff	Arizona	88	69
San Diego	California	77	60
Diego	New Mexico	80	60

Selecting Data

The Select statement is used to get data which matches the specified criteria. Here is the basic syntax:

ex)

```
select “columnname1”, “columnname2”
from “tablename”
where “condition”
```

Conditions Used In Where Clause

=	equals
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to
<>	not equal to

Like

Used to make complex searching easy. If you are trying to find all people’s names which begin with E for example:

ex)

```
select firstname from employee
where firstname LIKE 'E%';
```

Creating Tables

The statement to use is create table

Here is the syntax:

```
create table “tablename”
(“columnname”, “datatype”,
“columnname2”, “datatype”,
“columnname3”, “datatype”);
```

Creating Tables cont'd

Here is a real example:

```
create table employee
(first varchar(15),
last varchar(20),
age number(3),
address varchar(30),
city varchar(20),
state varchar(20));
```

Creating Tables - Steps

1. Use the command **create table**
2. Follow it with the correct table name
3. Put a parenthesis and type in the first column name
4. Follow it with the variable type (we will list them in a minute) then a comma
5. Continue the previous two steps until you have all your columns accounted for
6. Then put a parenthesis to close the columnname section and add a ; after it

Creating Tables - Rules

- Table and column names must start with a letter
- They can not exceed 30 characters
- They can not be key words such as create, insert, select, etc.

Creating Tables – Variables

- If you took algebra then y=2x might be familiar. y and x are unknown information, which is a variable.
- Now a string is a bunch of letters and numbers
- A number is a bunch of numbers
- A data type determines what a variable can hold, i.e. strings or numbers

Creating Tables – Data Types

- **char(size)** - all column entries must be = size, which you specify at the beginning, if size = 10 then you must have ten characters
- **varchar(size)** - all column entries must be less than or equal to whatever size is, if size is 10, then the string must be between 1-10 charcters

Creating Tables – Data Types cont'd

- **number(size)** - a number value that can not exceed, size columns, for example if you have size = 10, then you can only have 10 different digit places, like 1,000,000,000
- **date** - date value
- **number(size,d)** - This works the same as the regular number except d represents number of digits after the decimal.

Ex: decimal(*size*,*d*) *size* defaults to 18, if not specified. *d* defaults to 0, if not specified.

Creating Tables - Constraints

A constraint is a rule.

Some examples constraints are:

- unique - no two entries will be the same
- not null - no entry can be blank
- **primary key - unique identification of each row**
- primary keys will be very important to you as your knowledge of databases progresses

Inserting Information into Tables

To insert into tables you need only use the keyword insert. Here is the syntax:

```
insert into "tablename"
(“first_column”, ..., “last_column”)
values (“first_value”, ...,“last value”);
```

Inserting Information into Tables

Here is a practical example:

```
insert into employees
(first, last, age, address, city, state)
values ( ‘Rock', 'Duke', 45, '2130 Main Street',
'Hazard', 'Georgia');
```

Inserting Information into Tables Steps

****All strings should be enclosed by single quotes: 'string'****

1. Use the keyword "insert into" followed by the tablename
2. Then on the next line, in parenthesis, list all the columns you are inserting values for.
3. Then on the line after, type values, then in parenthesis, put the values in the same order as the columns they belong to

Updating Records

To update records use the "update" statement.

Here is the syntax:

update “tablename”
set “columnname” = “newvalue”,
“nextcolumn” = “newvalue2”, ...
where “columnname” OPERATOR
“value” and|or “columnname2 OPERATOR “value”

Updating Records cont’d

Here are some practical examples:

ex)

```
update phone_book
set area_code = 623
where prefix = 979;
```

This changes the area code all numbers beginning with 979 to 623

Updating Records cont’d

```
update phone_book
set last_name = 'Smith', prefix=555, sufix=9292
where last_name = 'Jones';
```

This changes everyone whose last name is Jones to Smith and their number to 555-9292

Deleting Records

Here is the syntax:

```
delete from “tablename”
where “columnname” OPERATOR “value”
and|or “columnname2” OPERATOR “value”
```

Deleting Records Examples

ex) **delete from employees;**

deletes all records from that table

ex) **delete from employee**

where lastname='May';

deletes all records for people whose last name is May

ex) **delete from employee**

where firstname='Mike' or firstname='Eric';

deletes all records for anyone whose first name is Mike or Eric

Deleting Tables

Use the drop command

```
drop table "tablename";
drop table employees;
```

Bye Bye Table, Hello Corporate Espoinage =)

Thank you

???