

(Very) Basic SQL

(2008.12.23)

Frank Ducrest

SQL: Structured Query Language

- “s-q-l”, “sequel”, “squwell”, “squeal”
- programming language for forming database queries
- mimics English
- statements are descriptions of what is wanted (functional programming), not the actions to take to get make something happen (procedural programming)

SQL Statement Categories

- **CREATE** – create a table
- **INSERT** – put data into a table
- **SELECT** – retrieve data
- **UPDATE** – change data
- **DELETE** – remove data
- **DROP** – delete a table

SQL Data Types

- **CHAR(*n*)** string *n* characters long
- **VARCHAR(*N*)** string up to *n* characters
- **INTEGER, INT** integer
- **DOUBLE** floating point
- **DATE** `java.sql.Date`

<http://java.sun.com/j2se/1.5.0/docs/api/java/sql/Date.html>

- **TIMESTAMP** `java.sql.Timestamp`

<http://java.sun.com/j2se/1.5.0/docs/api/java/sql/Timestamp.html>

- **BLOB** `java.sql.Blob` // binary large object

<http://java.sun.com/j2se/1.5.0/docs/api/java/sql/Blob.html>

- **Array** `java.sql.Array`

<http://java.sun.com/j2se/1.5.0/docs/api/java/sql/class-use/Array.html>

- **DECIMAL, NUMERIC** `java.math.BigDecimal`

<http://java.sun.com/j2se/1.5.0/docs/api/java/math/BigDecimal.html>

SQL Comparison Operators

- equals =
- greater than >
- less than <
- greater than or equal to >=
- less than or equal to <=
- not equal to <>
- pattern matching LIKE

note: '%' is anything (as '*' in UNIX / Linux)

- match set member IN

example:

WHERE age IN (18, 19, 20)

SQL Logical Operators

- **NOT**
- **AND**
- **OR**

Typical SQL Functions

- **COUNT(*)** - returns number of rows matching the WHERE condition
- **SUM(*col*)** - returns the sum of values in the column matching the WHERE condition
- **AVG(*col*)** – returns the average of values in the column matching the WHERE condition
- **MAX(*col*)** – returns the largest value in a column
- **MIN(*col*)** – returns the smallest value in a column

Syntax of Creating an Empty Table

CREATE TABLE *table-name* (

col-name *data-type*,

col-name *data-type*,

...

constraint

);

Examples of Creating Empty Tables

```
CREATE TABLE student (  
    id INTEGER PRIMARY KEY,  
    name VARCHAR(100),  
    major INTEGER  
);
```

Examples of Creating Empty Tables (continued)

```
CREATE TABLE schedule (  
    id INTEGER PRIMARY KEY,  
    classid INTEGER PRIMARY KEY  
);
```

```
CREATE TABLE classes (  
    classid INTEGER PRIMARY KEY,  
    classdesc CHARVAR(25)  
);
```

Syntax of Inserting into a Table

INSERT INTO table-name

(col-name-1, col-name-2, ...)

VALUES

(value1, value2, ...),

(value1, value2, ...),

...

);

Examples of Inserting into a Table

```
INSERT INTO student
```

```
    (id, name, major)
```

```
VALUES
```

```
    (09, 'Fred Flintstone', 15),
```

```
    (13, 'BarneyRubble', 08),
```

```
    (18, 'Betty Rubble', 34),
```

```
    (27, 'Wilma Flintstone', 23);
```

Examples of Inserting into a Table (continued)

```
INSERT INTO schedule
    (id, classid)
VALUES
    (09, 01),
    (09, 03),
    (09, 05),
    (13, 01),
    (13, 04);
```

```
INSERT INTO classes
    (classid, classdesc)
VALUES
    (01, 'Intro to Rocks'),
    (02, 'Intro to Stones'),
    (03, 'Intro to Gravel'),
    (04, 'Intro to Sand'),
    (05, 'Beginning Ice Age'),
    (06, 'Intro to Tracking'),
    (07, 'Beginning Fire');
```

Syntax of Retrieving Data

SELECT

col-name1, col-name-2, ...

*// use * for all col*

FROM

table-name-1, table-name-2, ...

WHERE *conditions*

// optional

ORDER BY *col-names*

// optional

;

// Note: can use “dot” notation to indicate one copy

// of an attribute that is used in two tables;

// example: *student.id*

Examples of Retrieving Data

```
SELECT
    name, major
FROM
    student
WHERE id = 9;
```

Returns: Fred Flintstone, 15

Examples of Retrieving Data (continued)

```
SELECT
    name, classdesc
FROM
    student, schedule, classes
WHERE
    student.id = 9
    AND student.id = schedule.id
    AND schedule.classid=classes.classid;
```

Returns:

- Fred Flintstone, Intro to Rocks**
- Fred Flintstone, Intro to Gravel**
- Fred Flintstone, Beginning Ice Age**

Syntax of Changing a Table

UPDATE *table-name*

SET *col-name1=value, col-name2=value, ...*

WHERE *condition* ;

Example of Changing a Table

```
UPDATE student
```

```
SET name='Fred E. Flintstone'
```

```
WHERE id=9;
```

Syntax of Deleting a Tuple

DELETE FROM *table-name*

WHERE *conditions* ;

Example of Deleting a Tuple

DELETE FROM schedule

WHERE id=9 AND classid=3;

Syntax of Deleting a Table

```
DROP TABLE table-name ;
```

Example of Deleting a Table

```
DROP TABLE schedule;
```