

# Data Manipulation Language (DML)

Lecture 03.06

*By Marina Barsky*

# Sub-sets of SQL

- Data Definition Language (DDL): CREATE, ALTER, DROP, RENAME
- Data Manipulation Language (DML): INSERT, UPDATE, DELETE
  - ✓ SELECT
- Transaction control: BEGIN; COMMIT; ROLLBACK;
- Data Control Language (DCL): GRANT, REVOKE

# RECAP: Data Definition Language (DDL)

```
CREATE TABLE table_name  
(  
    column_name1 data_type,  
    column_name2 data_type,  
    column_name3 data_type,  
    ....  
)
```

# Declaring primary keys

```
DROP TABLE IF EXISTS Movies;
```

```
DROP TABLE IF EXISTS Studios;
```

```
CREATE TABLE Studios (  
    name VARCHAR(20) PRIMARY KEY,  
    address VARCHAR(255)  
);
```

```
CREATE TABLE Movies (  
    title VARCHAR(20),  
    year INT,  
    length INT,  
    rating CHAR(2),  
    studioname VARCHAR(20),  
    PRIMARY KEY (title, year)  
);
```

# Altering, Dropping

```
ALTER TABLE Stars ADD [COLUMN] phone CHAR(16);
```

```
ALTER TABLE Stars ALTER COLUMN phone TYPE CHAR(26);
```

```
ALTER TABLE Stars DROP COLUMN phone;
```

```
DROP TABLE Stars;
```

```
DROP TABLE Movies;
```

```
DROP TABLE Studios;
```

# Getting information about tables

- Describe all tables:

`.tables`

- Describe table schema:

`.schema table_name`

# Sub-sets of SQL

- ✓ Data Definition Language (DDL): CREATE, ALTER, DROP, RENAME
- Data Manipulation Language (DML): INSERT, UPDATE, DELETE
  - ✓ SELECT
- Transaction control: BEGIN; COMMIT; ROLLBACK;
- Data Control Language (DCL): GRANT, REVOKE

# Data Modifications

- A modification command does not return a result as a query does, but it changes the database in some way.
- There are three kinds of modifications:
  1. *Insert* a tuple or tuples.
  2. *Delete* a tuple or tuples.
  3. *Update* the value(s) of an existing tuple or tuples.



# Insertion

- To insert a single tuple:

```
INSERT INTO <relation>  
VALUES ( <list of values> );
```

## Example

- Consider **MovieExec**(name, address, cert#, netWorth)

```
INSERT INTO MovieExec  
VALUES('Melanie Griffith', '34 Boston Blvd', 700, 300000);
```

# Specifying Attributes in INSERT

- We may add to the relation name a list of attributes.

```
INSERT INTO MovieExec(name, address, cert, netWorth)  
VALUES('Melanie Griffith', NULL, 700, 3000000);
```

- There are two reasons to do so:
  1. We forget the standard order of attributes for the relation.
  2. We don't have values for all attributes.

# Inserting Many Tuples

- We may insert the entire result of a query into a relation, using the form:

```
INSERT INTO <relation>  
( <subquery> );
```

## Example

```
CREATE TABLE DisneyMovies(  
    name VARCHAR2(25),  
    year INT  
);
```

```
INSERT INTO DisneyMovies  
(SELECT title, year  
FROM Movie  
WHERE studioName = 'Disney'  
);
```

# Deletion

- To delete tuples satisfying a condition from some relation:

```
DELETE FROM <relation>  
WHERE <condition>;
```

## Example

- Delete from the **Movie** table all the Disney's movies:

```
DELETE FROM Movie  
WHERE studioName ='Disney';
```

# Example: Delete all Tuples

- Make the relation Movie empty:

**DELETE FROM Movie;**

- No WHERE clause needed here.

# Updates

- To change certain attributes in certain tuples of a relation:

UPDATE <relation>

SET <list of attribute assignments>

WHERE <condition on tuples>;

## Example

- Change the length of 'Godzilla' to 200.

UPDATE Movie

SET length = 200

WHERE title = 'Godzilla';

# Another example

Suppose that Brown's movies have approximately 20 min of info before starting.

So, let's take that 20 min off.

```
UPDATE Movie
SET length = length - 20
WHERE (title, year) IN
    (SELECT title, year
     FROM Movie, Movieexec
     WHERE Movie.producerc = Movieexec.cert
           AND name = 'Brown');
```