DML – Data Manipulation Language

Data Manipulation Language - DML

- Language used to query, insert, update and delete data within a database
- Commands include:
  - SELECT
  - **INSERT INTO**
  - UPDATE
  - DELETE

Step 3 – Inserting Data into Tables

- The INSERT command is used to insert data into tables
- Can be:
  - One row at a time
  - Multiple rows
  - Using data from other tables

- **Insert – Syntax 1** (in all columns - no column listing required)

```
INSERT INTO tbl_name VALUES (val1,val2,...)
```

Staff table:

StaffID	StaffName	DateOfBirth	Salary
1	Buffy Summers	1987-09-15	27000.00

- Example:

```
INSERT INTO staff VALUES (NULL, 'Buffy Summers', '1987-09-15', 27000);
```

  - '**NULL**' here is a dummy place holder for the AUTONUMBER

Step 3 – Inserting Data into Tables

- The INSERT command is used to insert data into tables
- Can be:
  - One row at a time
  - Multiple rows
  - Using data from other tables

- **Insert – Syntax 1** (in all columns - no column listing required)

```
INSERT INTO tbl_name VALUES (val1,val2,...)
```

Staff table:

StaffID	StaffName	DateOfBirth	Salary
1	Buffy Summers	1987-09-15	27000.00
2	Buffy Summers	1987-09-15	27000.00

- Example:

```
INSERT INTO staff VALUES (NULL, 'Buffy Summers', '1987-09-15', 27000);
```

  - '**NULL**' here is a dummy place holder for the AUTONUMBER

Step 3 – Inserting Data into Tables

- **Insert – Syntax 2** (specific columns - column listing required)

```
INSERT INTO tbl_name (col1,col2,...)
                VALUES  (val1,val2,...);
```

- Example:

```
INSERT INTO Staff(StaffName, DateOfBirth, Salary) VALUES ('Teddy Bear', '1983-12-03', 87125.02);
```

- Note that the above insertion statement did not include StaffID: DBMS automatically adds next available value because StaffID is AUTO_INCREMENT.

Staff table:

StaffID	StaffName	DateOfBirth	Salary
1	Buffy Summers	1987-09-15	27000.00
2	Buffy Summers	1987-09-15	27000.00
3	Teddy Bear	1983-12-03	87125.02

Step 3 – Inserting Data into Tables

Write SQL statement to insert the following employees:

- John Smith born on the 20th September, 1972 and has an annual salary of $25,000

  INSERT INTO STAFF VALUES (NULL,'John Smith', '1972-9-20', 25000);

- Jane Doe born on the 25th January, 1969 and has an annual salary of $55,000

  INSERT INTO STAFF VALUES (NULL,'Jane Doe', '1969-1-25', 55000);

- Jack Jones born on the 19th October, 1984 and has an annual salary of $35,000

  INSERT INTO STAFF VALUES (NULL,'Jacek Jones', '1984-10-19', 35000);

Staff table:

StaffID	StaffName	DateOfBirth	Salary
1	Buffy Summers	1987-09-15	27000.00
2	Buffy Summers	1987-09-15	27000.00
3	Teddy Bear	1983-12-03	87125.02
4	John Smith	1972-09-20	25000.00
5	Jane Doe	1969-01-25	55000.00
6	Jacek Jones	1984-10-19	35000.00

Step 3 – Inserting Data into Tables

- Write SQL statement to insert information about the Sales department, which is managed by Jane Doe full-time, which employs Jacek Jones for half of this time, and which has an annual budget of $500,000

```
INSERT INTO DEPARTMENT
      VALUES (1, 'Sales', 500000, 2);
```

Department table:

DepartmentID	DepartmentName	Budget	ManagerID
1	Sales	500000	2

```
INSERT INTO WORKALLOCATION
      VALUES (2, 1, 1);

INSERT INTO WORKALLOCATION
      VALUES (3, 1, 0.5);
```

WorkAllocation table:

StaffID	DepartmentID	PercentageTime
2	1	1
3	1	0.5

Step 3 – Inserting Data into Tables

- **Insert – Syntax 3:** Insert multiple rows at a time

- **Example:**

```
INSERT INTO Staff (StaffName, DateOfBirth, Salary) VALUES
('Teddy Bear', '1983-12-03', 87125.02),
('Fred Smith', '1956-06-30', 25125.02);
```

Staff table:

StaffID	StaffName	DateOfBirth	Salary
1	Buffy Summers	1987-09-15	27000.00
2	Buffy Summers	1987-09-15	27000.00
3	Teddy Bear	1983-12-03	87125.02
4	John Smith	1972-09-20	25000.00
5	Jane Doe	1969-01-25	55000.00
6	Jacek Jones	1984-10-19	35000.00
7	Teddy Bear	1983-12-03	87125.02
8	Fred Smith	1956-06-30	25125.02

Step 3 – Inserting Data into Tables

- **Insert – Syntax 4:** Insert data from other tables, usually multiple rows at a time!
- Insert many rows into an existing table from one or more tables
- Example:

```
INSERT INTO Staff2
SELECT *
FROM Staff;
```

- **Create Staff2 table first!**

```
CREATE TABLE staff2 (
StaffId int( 11 ) NOT NULL,
StaffName varchar( 30 ) DEFAULT NULL ,
DateOfBirth date DEFAULT NULL ,
Salary decimal( 10, 2 ) DEFAULT NULL ,
PRIMARY KEY ( StaffId )
) ENGINE = InnoDB;
```

- Note: Staff2 should have the exact value types of the Staff. For AUTONUMBER the type in Staff2 should be an INT

- Create backup table:

```
CREATE TABLE Staff3
SELECT * FROM Staff;
```

Staff 3 table:

+ Options

StaffID	StaffName	DateOfBirth	Salary
1	Buffy Summers	1987-09-15	27000.00
2	Buffy Summers	1987-09-15	27000.00
3	Teddy Bear	1983-12-03	87125.02
4	John Smith	1972-09-20	25000.00
5	Jane Doe	1969-01-25	55000.00
6	Jacek Jones	1984-10-19	35000.00
7	Teddy Bear	1983-12-03	87125.02
8	Fred Smith	1956-06-30	25125.02

Staff 2 table:

StaffId	StaffName	DateOfBirth	Salary
1	Buffy Summers	1987-09-15	27000.00
2	Buffy Summers	1987-09-15	27000.00
3	Teddy Bear	1983-12-03	87125.02
4	John Smith	1972-09-20	25000.00
5	Jane Doe	1969-01-25	55000.00
6	Jacek Jones	1984-10-19	35000.00
7	Teddy Bear	1983-12-03	87125.02
8	Fred Smith	1956-06-30	25125.02

Step 3 – Inserting Data into Tables

Insert rules:

- If you specify the column list every column in the table must be provided with values in the VALUES() list or by the SELECT
- If you do not specify the list of all the columns in the table, unnamed columns are set to their default values (e.g. NULL, 'N', 0).
- If you do not specify the column list at all, every column in the table must be provided with values in the VALUES() list or by the SELECT
  - E.g. INSERT INTO STAFF VALUES (NULL,'John Smith', 26, 25000);

Thank you