Database Systems

Chapter 10
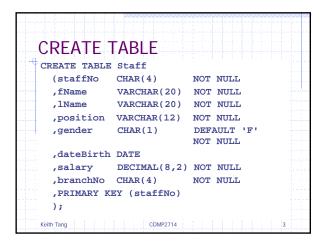DML – Insert, Update, Delete

---

Topics

◆ Manipulate data in database tables
◆ INSERT
  ▪ Single-row INSERT
  ▪ Multi-row INSERT
◆ UPDATE
  ▪ WHERE clause: selected rows vs all rows
◆ DELETE
  ▪ WHERE clause: selected rows vs all rows
◆ MERGE

CREATE TABLE

```
CREATE TABLE Staff
  (staffNo    CHAR(4)      NOT NULL
  ,fName      VARCHAR(20)  NOT NULL
  ,lName      VARCHAR(20)  NOT NULL
  ,position   VARCHAR(12)  NOT NULL
  ,gender     CHAR(1)      DEFAULT 'F'
                          NOT NULL
  ,dateBirth  DATE
  ,salary     DECIMAL(8,2) NOT NULL
  ,branchNo   CHAR(4)      NOT NULL
  ,PRIMARY KEY (staffNo)
  );
```

INSERT ... VALUES

◆ Insert a new row into Staff table supplying data for all columns

◆ ```
INSERT INTO Staff
VALUES
('SG16', 'Alan', 'Brown',
 'Assistant', 'M', DATE'1957-05-25',
 8300, 'B003');
```

---

## INSERT ... VALUES

◆ ```
INSERT INTO TblName [(columnList)]
VALUES (dataValueList);
```

◆ dataValueList (comma separated) must match columnList (comma separated) as follows:
- number of items in each list must be same
- must be direct correspondence in position of items in two lists
- data type of each item in dataValueList must be compatible with data type of corresponding column

---

INSERT ... VALUES

◆ ```
INSERT INTO TblName [(columnList)]
VALUES (dataValueList);
```

◆ columnList is optional; if omitted, SQL assumes a list of all columns in their original CREATE TABLE order
◆ Any columns omitted must have been declared as NULL when table was created, unless DEFAULT was specified when creating column

## INSERT using DEFAULT & NULL

◆ Insert a new row into Staff table supplying data for all mandatory columns

```
INSERT INTO Staff (fName, lName,
    position, salary, branchNo, staffNo)
VALUES ('Anne', 'Jones',
    'Assistant', 8100, 'B003', 'SG44' );
```

◆ Or

```
INSERT INTO Staff
VALUES ('SG44', 'Anne', 'Jones',
    'Assistant', DEFAULT, NULL,
    8100, 'B003');
```

INSERT ... SELECT

◆Second form of INSERT allows multiple rows to be **copied** from one or more tables to another:

```
INSERT INTO TblName[(columnList)]
    SELECT ...
    ;
```

INSERT ... SELECT

```
CREATE TABLE RetiredStaff
  (staffNo   CHAR(4)       NOT NULL
  ,fName     VARCHAR(20)   NOT NULL
  ,lName     VARCHAR(20)   NOT NULL
  ,position  VARCHAR(12)   NOT NULL
  ,gender    CHAR(1)       NOT NULL
  ,dateBirth DATE
  ,salary    DECIMAL(8,2)  NOT NULL
  ,branchNo  CHAR(4)       NOT NULL
  ,PRIMARY KEY (staffNo)
  );
```

INSERT ... SELECT

```
INSERT INTO RetiredStaff
   SELECT *
   FROM Staff
   WHERE dateBirth IS NOT NULL
     AND dateBirth < TO_DATE('1938-01-01');


DELETE FROM Staff
   WHERE dateBirth IS NOT NULL
     AND dateBirth < TO_DATE('1938-01-01');

-- Note Oracle's TO_DATE built-in function
-- Or use SQL standard: DATE'1938-01-01'
```

UPDATE

```
UPDATE TableName
SET columnName1 = dataValue1
   [,columnName2 = dataValue2...]
[WHERE searchCondition];
```

◈ TableName can be name of a base table or an updatable view
◈ SET clause specifies names of one or more columns that are to be updated

UPDATE

◈ WHERE clause is optional:
  ▪ if omitted, named columns are updated for all rows in table
  ▪ if specified, only those rows that satisfy searchCondition are updated
◈ How many/**which columns** are required to be specified to limit change to just 1 row?
◈ New dataValue(s) must be compatible with data type for corresponding column

UPDATE

- Give all staff a 3% pay increase:
- **-- Updating all rows**
- **UPDATE Staff**
  SET salary = salary*1.03;
- Give all Managers a 5% pay increase:
- **-- Updating only selected rows**
- **UPDATE Staff**
  SET salary = salary*(1+0.05)
  WHERE position = 'Manager';

- **What is wrong here? No error msg!**

UPDATE Multiple Columns

- Promote David Ford (staffNo = 'SG14') to Manager and change his salary to 58,000

- **UPDATE Staff**
  SET position = 'Manager'
  ,salary   = 58000
  WHERE staffNo = 'SG14';

DELETE

```
DELETE
FROM TableName
[WHERE searchCondition];
```

- TableName can be name of a base table or an updatable view
- searchCondition is optional;
  - if omitted, all rows are deleted from table, but this does not delete table.
  - If search_condition is specified, only those rows that satisfy condition are deleted.

DELETE Specific Rows

◆ Delete all staff members who have retired and their data archived:

◆
```
DELETE
FROM Staff
WHERE dateBirth IS NOT NULL
  AND dateBirth <
        TO_DATE('1938-01-01');
```

◆ Delete all records from the Staff table:

◆
```
DELETE
FROM Staff;
```