▨ SQL (SKS)

- **DML Statements**

— How to store data?

```
DEPT(DCODE, DNAME)
STUDENT(ROLL, NAME, PH_NO, DT_BTH, DCODE).
```

```
SQL>INSERT INTO STUDENT (attr.n values inserted in order)
     VALUE(1, 'ABC', 975624, '02-FEB-2002', 'D1');
```

All attr ⇒ ordered as in table creation

```
SQL> INSERT INTO STUDENT(ROLL, NAME, DCODE)
      VALUES(2, 'XYZ', 'D2');
```

- either all attr. vals given in order or subset of them mentioning the names with one-to-one mapping

→ For the missing attr. either null placed or default val.

— How to retrieve?

→ List of expns.

```
SQL > SELECT ROLL NAME
      FROM STUDENT;
```

```
SQL> SELECT *
     FROM STUDENT
     WHERE DCODE = 'D1'; (AND, OR, NOT can be used).
```

checked with each tuple

```
SQL> SELECT *
     FROM STUDENT
     WHERE DCODE IN ('D1', 'D2'...)
```

List of values

DCODE = 'D1' OR DCODE = 'D2' OR DCODE = 'D3'

~~SQL>~~

```
DEPT(DCODE, DNAME)
SUBJECT(SCODE, SNAME)
STUDENT(ROLL, NAME, DT_BTH, DCODE)
RESULT(ROLL, SCODE, SCORE)
```

All students who scored b/w 50 & 80

```
SQL> SELECT ROLL
     FROM RESULT
     WHERE SCODE = 'S1' AND SCORE >= 50 AND SCORE <= 80
```

↓ alternate

SCORE BETWEEN 50 AND 80

IN (list of columns)

True if equality holds with any of the values in the list

ANY (list of cols) → if condn. holds for any tuple

ALL (list of cols) → " " all tuples

→ unlike IN, here condn. is not fixed

e.g. >= ANY (...) → if greater than any col. considered

>= ALL (...) → if greater than all

* bad examples b/c ...

LIKE 'pattern string'

```
SELECT *
FROM STUDENT
WHERE NAME LIKE 'SA%';
```

(negation using NOT LIKE instead)

IS NULL

IS NOT NULL

→ to check if certain value is null

— How to modify data?

```
SQL> UPDATE RESULT
     SET SCORE = SCORE - 5
     WHERE SCODE = 'DBMS';
```

```
SQL> DELETE FROM RESULT
     WHERE _____ ;
```

→ tuples dropped, schema remains

(deletes only those tuples that satisfy condn.)

- Built-in Functions

Single Row Func.

=) Acts on every single tuple & provides an output

Aggregate Func.

→) acts on a collection of tuples & provides one result for the collection

→ ~~AB~~ NUMERIC

ABS(num.)

CEIL(num.)

FLOOR(num.)

MOD($\overset{\text{int}}{m}$, $\overset{\text{int}}{n}$)

POW(m, n)

LOG( )

EXP( )

SQRT( )

ROUND($\overset{\text{number}}{m}$, n) → upto which place of decimal rounding occurs

(100.753, 1) → 100.8

(100.75, 0) → 101

(189.72, -1) → 190

→ CHAR/STRING

- UPPER( )
- LOWER( )
- INIT CAP( ) → 1st char made capital, rest made small
- CHAR(ascii val)
- ASCII(char)
- LTRIM( )
- RTRIM( )
- LENGTH( )

— Aggregate Funcs

- COUNT(Attr).

```
SELECT COUNT(DCODE)
FROM STUDENT
WHERE ——— ;
```

→ in how many tuple of the collection, DCODE is not null

if COUNT (DISTINCT DCODE) → no. of unique non-null DCODE

COUNT(*) → no. of tuples counted that satisfy any condn. or just present in the table

- SUM()

```
SELECT SUM(SCORE)
FROM RESULT
WHERE ROLL = 5 ;
```

AVG()
MAX()
MIN()
STDDEV()
VAR()

} null vals if present will cause prob. & thus need to be filtered

How to change schema?
EXISTS operator