BASIC SQL

```
CREATE TABLE <table_name> (
    <column>  <type>,
    ...
);
```

```
INSERT INTO <table_name>
VALUES (...), (...), ..., (...);
```

↳ can use DEFAULT keyword in INSERT

↳ Attributes w/o DEFAULT in CREATE TABLE have default value NULL

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

```
UPDATE <table>
SET <assignments>
WHERE <condition>;
```

```
ORDER BY <column> ASC/DESC
```

```
CAST ( term AS <type>)
```

↳ CAST (102.4675 AS NUMERIC(5,2)) = 102.47

↳ AVG (CAST (term AS NUMERIC(p,s))

```
CASE WHEN <bool-expr>
     THEN <value-expr>
     ELSE <value-expr>
END
```

term LIKE pattern
- _ wildcard matching any one char
- % wildcard matching any substr.

```
SELECT *
FROM Customer
WHERE name LIKE 'K___%';
```
case-sensitive!

RELATIONAL ALGEBRA

$$\sigma_{\theta_1}(\sigma_{\theta_2}(R)) \equiv \underbrace{\sigma_{\theta_1 \wedge \theta_2}(R)}_{\text{Faster}}$$

$$\underbrace{\sigma_{\theta}(\pi_a(R))}_{\text{keeps less stuff in main memory}} \equiv \pi_a(\sigma_{\theta}(R))$$

CARTESIAN PDT.

row – card(R × S) = card(R) × card(S)

col. – arity(R × S) = arity(R) + arity(S)

NATURAL JOIN

Customer: (CustID), Name

Account: Number, (CustID)

$$\text{Customer} \bowtie \text{Account} \equiv \pi_{X \cup Y}\left(\sigma_{\text{CustID} = \text{CustID}'}\left(\text{Customer} \times \rho_{\text{CustID} \to \text{CustID}'}(\text{Account})\right)\right)$$

X: all attr. of Customer; Y: all attr. of Account

Primitive Operations — $\pi, \sigma, \times, \rho, \cup, -$

Derived Operations —

- $\bowtie$ can be expressed in terms of $\pi, \sigma, \times, \rho$
- $\cap$ can be expressed in terms of $-$

$(R \cap S \equiv R - (R - S))$

$$R \bowtie_\theta S \equiv \sigma_\theta (R \times S)$$

$$R \ltimes_\theta S \equiv \pi_X (R \bowtie_\theta S)$$

X: attr. of R

$$R \bar{\ltimes}_\theta S \equiv R - (R \ltimes_\theta S)$$

DIVISION

R over set of attributes X

S over set of attributes $Y \subset X$

Let $Z = X - Y$

$$R \div S = \pi_Z(R) - \pi_Z(\pi_Z(R) \times S - R)$$