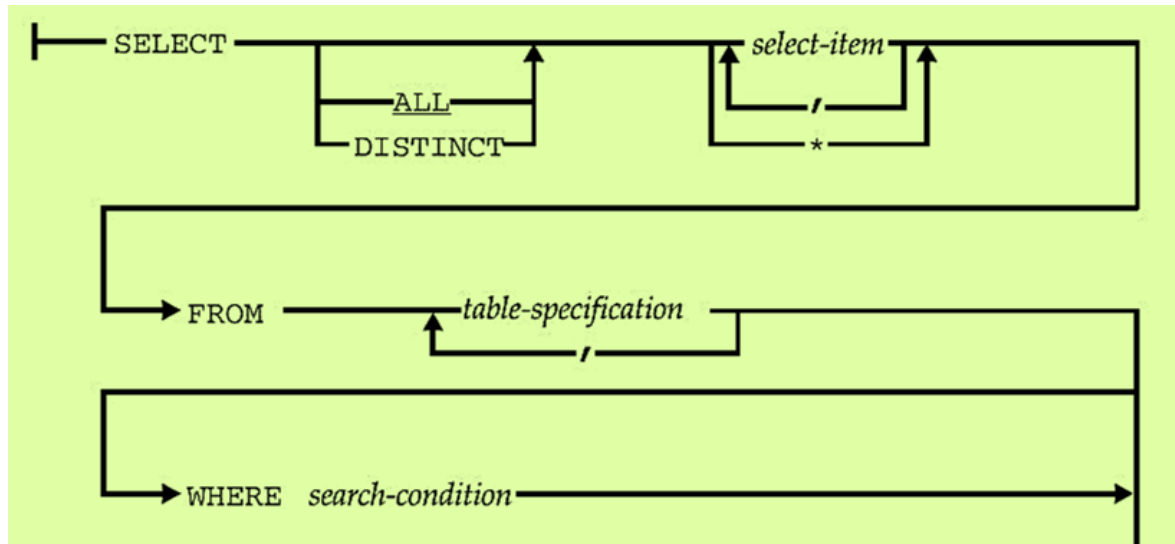


# Simple Queries

September, 2023

# Syntax of SQL SELECT Statement



Sample Database

Order Processing Application

- The **CUSTOMER** table (Stores data about each customer)  
(CUST\_NUM, COMPANY, CUST\_REP, CREDIT\_LIMIT)  
- 2111, JCP Inc, 103, 50,000
- The **SALESREPS** table (Data about salesperson)  
(EMPL\_NUM, NAME, AGE, REP\_OFFICE, TITLE, HIRE\_DATE, MANAGER, QUOTA, SALES)  
- 105, Bina Singh, 37, 13, Sales Rep, 12-Feb-88, 105, 350000, 367911
- The **OFFICES** table (Data about each of the 5 sales offices)  
(OFFICE, CITY, REGION, MGR, TARGET, SALES)  
- 13, Kanpur, Eastern, 105, 350000, 367911
- The **ORDERS** table (Every order placed by a customer, identifying the salesperson who took the order, the product ordered, the quantity and amount of the order)  
(ORDER\_NUM, ORDER\_DATE, CUST, REP, MFR, PRODUCT, QTY, AMOUNT)  
- 11269, 17-Dec-89, 2111, 103, ACI, 2A44L, 731000
- The **PRODUCT** table (Stores data about each product available)  
(MFR\_ID, PRODUCT\_ID, DESCRIPTION, PRICE, QTY\_ON\_HAND)  
- ACI, 2A44L, Plate, 355.00, 38

# Simple Queries

- List the names, offices, and hire dates of all salespeople.
- List the location, region, and sales of each sales office.
- List the city, region, and amount over/under target for each office.
- Show the value of inventory for each product.
- Show me the result if I raised each salesperson's quota by 3% of their year to date sales.
- List the sales for each city.
- Show me all the data in the OFFICES table.
- List the employee nos. of all sales office managers.

# Simple Queries

- List the names, offices, and hire dates of all salespeople.

```
SELECT NAME, REP_OFFICE, HIRE_DATE  
FROM SALESREPS;
```

NAME	REP_OFFICE	HIRE_DATE
Bill Adams	13	2006-02-12
Mary Jones	11	2007-10-12
Sue Smith	21	2004-12-10
Sam Clark	11	2006-06-14
Bob Smith	12	2005-05-19
Dan Roberts	12	2004-10-20
Tom Snyder	NULL	2008-01-13
Larry Fitch	21	2007-10-12
Paul Cruz	12	2005-03-01
Nancy Angelli	22	2006-11-14

# Simple Queries

- List the location, region, and sales of each sales office.

```
SELECT CITY, REGION, SALES  
FROM OFFICES;
```

CITY	REGION	SALES
-----	-----	-----
Denver	Western	\$186,042.00
New York	Eastern	\$692,637.00
Chicago	Eastern	\$735,042.00
Atlanta	Eastern	\$367,911.00
Los Angeles	Western	\$835,915.00

- List the city, region, and amount over/under target for each office.

```
SELECT CITY, REGION, (SALES - TARGET)  
FROM OFFICES;
```

# Simple Queries

- Show the value of inventory for each product.

```
SELECT MFR_ID, PRODUCT_ID, DESCRIPTION, (QTY_ON_HAND * PRICE)
FROM PRODUCTS;
```

- Show me the result if I raised each salesperson's quota by 3% of their year to date sales.

```
SELECT NAME, QUOTA, (QUOTA + (.03*SALES))
FROM SALESREPS;
```

NAME	QUOTA	(QUOTA+ (.03*SALES))
-----	-----	-----
Bill Adams	\$350,000.00	\$361,037.33
Mary Jones	\$300,000.00	\$311,781.75
Sue Smith	\$350,000.00	\$364,221.50
Sam Clark	\$275,000.00	\$283,997.36
Bob Smith	\$200,000.00	\$204,277.82
Dan Roberts	\$300,000.00	\$309,170.19
Tom Snyder	NULL	NULL



# Simple Queries

- List the sales for each city.

```
SELECT CITY, 'has sales of', SALES  
FROM OFFICES;
```

- Show me all the data in the OFFICES table.

```
SELECT *  
FROM OFFICES;
```

- List the employee nos. of all sales office managers.

```
SELECT DISTINCT MGR  
FROM OFFICES;
```

```
SELECT MGR  
FROM OFFICES;
```

```
MGR  
----  
108  
106  
104  
105  
108
```

# Row Selection

- What are the name, quota, and sales of employee no. 107?

```
SELECT NAME, QUOTA, SALES
FROM SALESREPS
WHERE EMPL_NUM = 107;
```

- List the name and hire date of anyone with sales over Rs. 50,000.

```
SELECT NAME, HIRE_DATE
FROM SALESREPS
WHERE SALES > 500000.00;
```

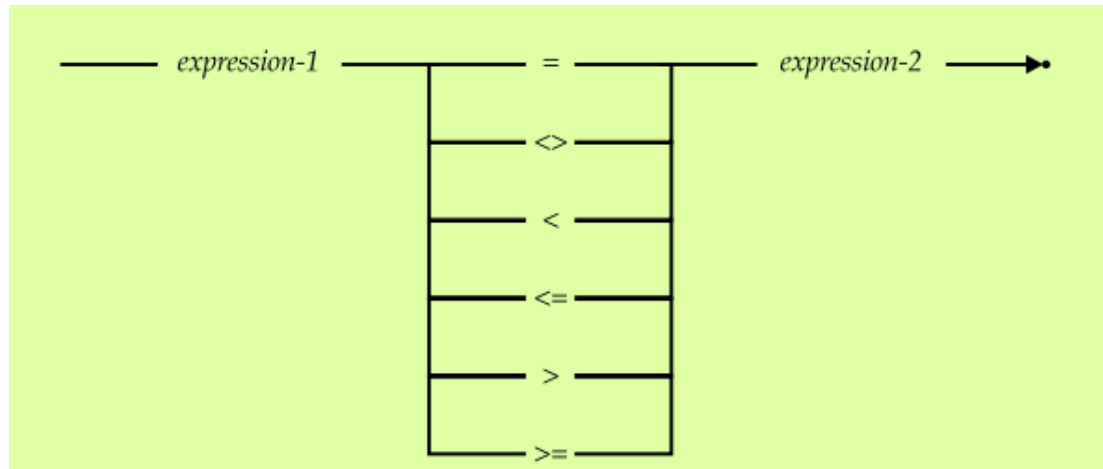
NAME	HIRE_DATE
-----	-----

# Search Conditions

- Comparison
- Range test
- Set membership test
- Pattern matching test
- Null value test

# Comparison Test

(=, <>, <, <=, >, >=)



- Offices whose sales fall below 80% of target.

```
SELECT CITY, SALES, TARGET
FROM OFFICES
WHERE SALES < (.8 * TARGET) ;
```

# Comparison Test

(=, <>, <, <=, >, >=)

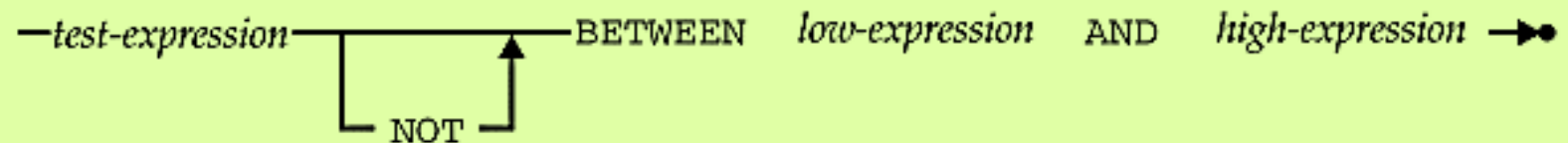
- List the offices not managed by emp. No. 108.

```
SELECT CITY, MGR  
FROM OFFICES  
WHERE MGR <> 108;
```

- List salespeople who are over quota.

```
SELECT NAME  
FROM SALESREPS  
WHERE SALES > QUOTA;
```

# Range test



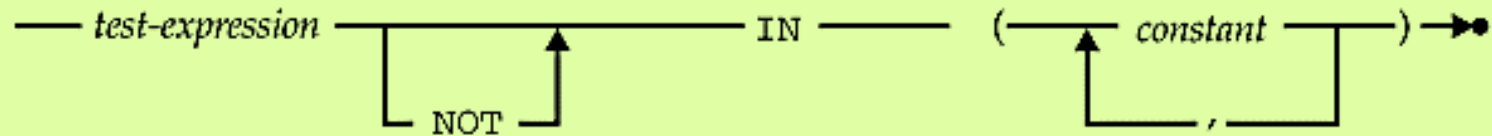
- Find orders placed in the last quarter of 2007.

```
SELECT ORDER_NUM, ORDER_DATE, MFR, PRODUCT, AMOUNT
FROM ORDERS
WHERE ORDER_DATE BETWEEN '2007-10-01' AND '2007-12-31';
```

- List salesperson whose sales are not in the range of 80% and 120% of quota.

```
SELECT NAME, SALES, QUOTA
FROM SALESREPS
WHERE SALES NOT BETWEEN (.8 * QUOTA) AND (1.2 * QUOTA);
```

# Set membership test



- List salespeople working in Delhi, Mumbai or Kolkata.

```
SELECT NAME, QUOTA, SALES
FROM SALESREPS
WHERE REP_OFFICE IN (11, 13, 22);
```

- Find all orders placed on a Friday in Jan 2008.

```
SELECT ORDER_NUM, ORDER_DATE, AMOUNT
FROM ORDERS
WHERE ORDER_DATE IN ('2008-01-04', '2008-01-11',
                    '2008-01-18', '2008-01-25');
```

# Pattern Matching Test (LIKE)

- Show the credit limit for Smithson Corp.

```
SELECT COMPANY, CREDIT_LIMIT  
FROM CUSTOMERS  
WHERE COMPANY = 'Smithson Corp.';
```

- Wildcard Characters (% and \_)

```
SELECT COMPANY, CREDIT_LIMIT  
FROM CUSTOMERS  
WHERE COMPANY LIKE 'Smith% Corp.';
```

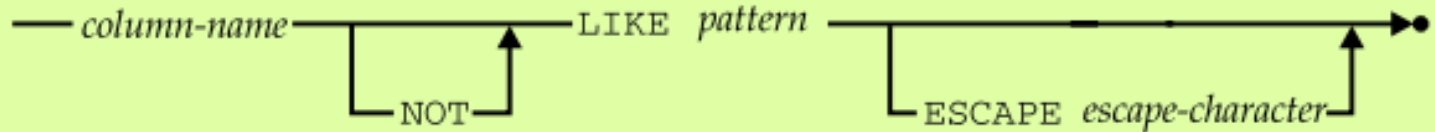
Any of the following names would match the pattern:

Smith Corp.      Smithson Corp.      Smithsen Corp.      Smithsonian Corp.

but these names would not: SmithCorp      Smithson Inc.



# Pattern Matching Test (LIKE)



- Wildcard Character - Underscore

```
SELECT COMPANY, CREDIT_LIMIT  
FROM CUSTOMERS  
WHERE COMPANY LIKE 'Smiths_n Corp.';
```

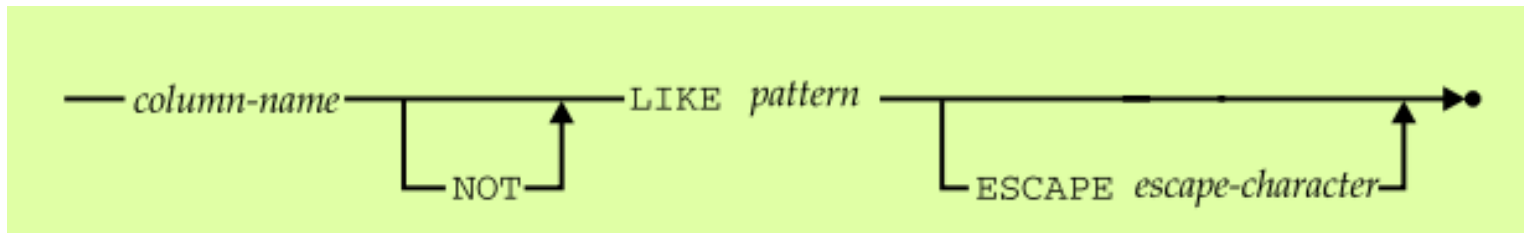
Any of these names will match the pattern:

Smithson Corp. Smithsen Corp. Smithsun Corp.

but these names will not: Smithsoon Corp. Smithsn Corp.

```
SELECT COMPANY, CREDIT_LIMIT  
FROM CUSTOMERS  
WHERE COMPANY LIKE 'Smiths_n %';
```

# Pattern Matching Test (LIKE)



- Escape Characters
- How to match the wildcard characters themselves as literal character
- Find products whose product IDs start with the four letters “A%BC”.

```
SELECT ORDER_NUM, PRODUCT
FROM ORDERS
WHERE PRODUCT LIKE 'A$%BC%' ESCAPE '$';
```

The first percent sign in the pattern, which follows an escape character, is treated as a literal percent sign; the second functions as a wildcard.

# Null Value Test

- To check explicitly for NULL values in a search condition
- Find the salespersons not yet assigned to an office.

```
SELECT NAME  
  FROM SALESREPS  
 WHERE REP_OFFICE = NULL;
```

```
SELECT NAME  
  FROM SALESREPS  
 WHERE REP_OFFICE IS NULL;
```

- List the salespeople who have been assigned to an office.

```
SELECT NAME  
  FROM SALESREPS  
 WHERE REP_OFFICE IS NOT NULL;
```

# Compound Search Conditions

- Find salespeople who are under quota or with sales under Rs. 300,000.

```
SELECT NAME, QUOTA, SALES
FROM SALESREPS
WHERE SALES < QUOTA
      OR SALES < 300000.00;
```

AND	TRUE	FALSE	NULL
TRUE	TRUE	FALSE	NULL
FALSE	FALSE	FALSE	FALSE
NULL	NULL	FALSE	NULL

OR	TRUE	FALSE	NULL
TRUE	TRUE	TRUE	TRUE
FALSE	TRUE	FALSE	NULL
NULL	TRUE	NULL	NULL

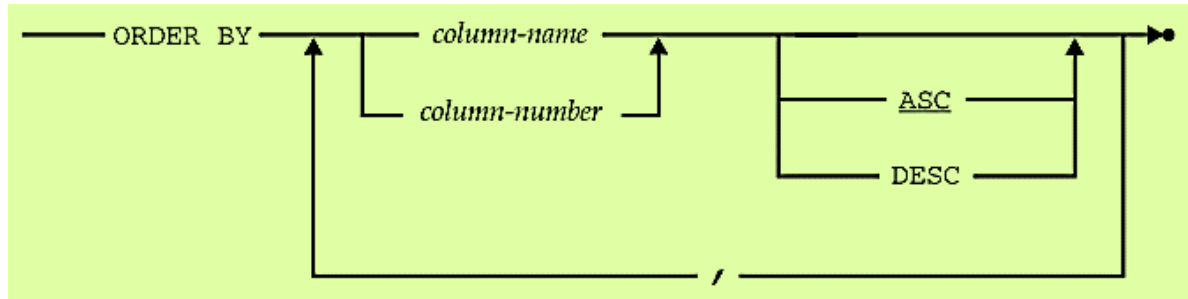
NOT	TRUE	FALSE	NULL
	FALSE	TRUE	NULL

# Compound Search Conditions

- Find all salespeople who either
  - Work in office 22, 11, or 12; or
  - Have no manager and were hired since June 2006; or
  - Are over quota, but have sales of Rs.600,000, or less.

```
SELECT NAME
FROM SALESREPS
WHERE (REP_OFFICE IN (22, 11, 12))
      OR (MANAGER IS NULL AND HIRE_DATE >= '2006-06-01')
      OR (SALES > QUOTA AND NOT SALES > 600000.00);
```

# Sorting Query Results



- Show the sales of each office, sorted in alphabetical order by region, and within each region by city.

```
SELECT CITY, REGION, SALES
FROM OFFICES
ORDER BY REGION, CITY;
```

- Show the sales of each office in the decreasing order of sales made by these offices.

```
SELECT CITY, REGION, SALES
FROM OFFICES
ORDER BY SALES DESC;
```

# Rules for Single-Table Query Processing

1. FROM clause.
2. WHERE clause
3. SELECT clause
4. SELECT DISTINCT
5. ORDER BY clause

# Combining Query Results

- List all the products where the price of product exceeds Rs.2,000 or where more than Rs. 30,000 of the product has been ordered.

PRODUCTS Table


PRICE > \$2,000

MFR_ID	PRODUCT_ID
ACI	4100Y
REI	2A44L
ACI	4100Z
REI	2A44R

ORDERS Table


MFR	PRODUCT
REI	2A44L
REI	2A44R
IMM	775C

AMOUNT > \$30,000



# Combining query results (UNION)

- List all the products where the price of product exceeds Rs.2,000 or where more than Rs. 30,000 of the product has been ordered.

```
SELECT MFR_ID, PRODUCT_ID  
FROM PRODUCTS  
WHERE PRICE > 2000.00;
```

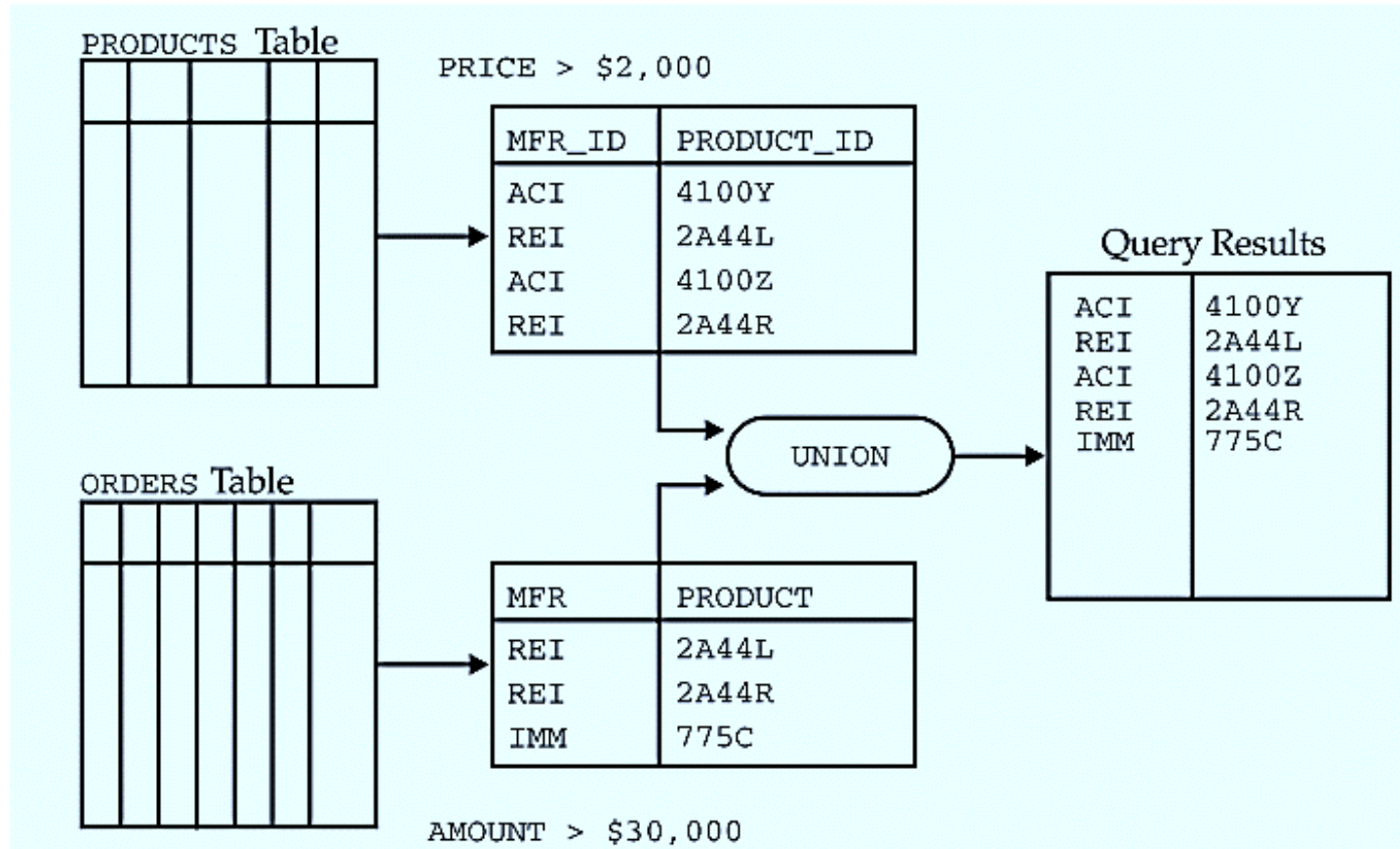
UNION

```
SELECT DISTINCT MFR, PRODUCT  
FROM ORDERS  
WHERE AMOUNT > 30000.00;
```

- Restrictions

# Combining Query Results

- Unions and Duplicate Rows



# Combining query results (UNION)

- Unions and Duplicate Rows

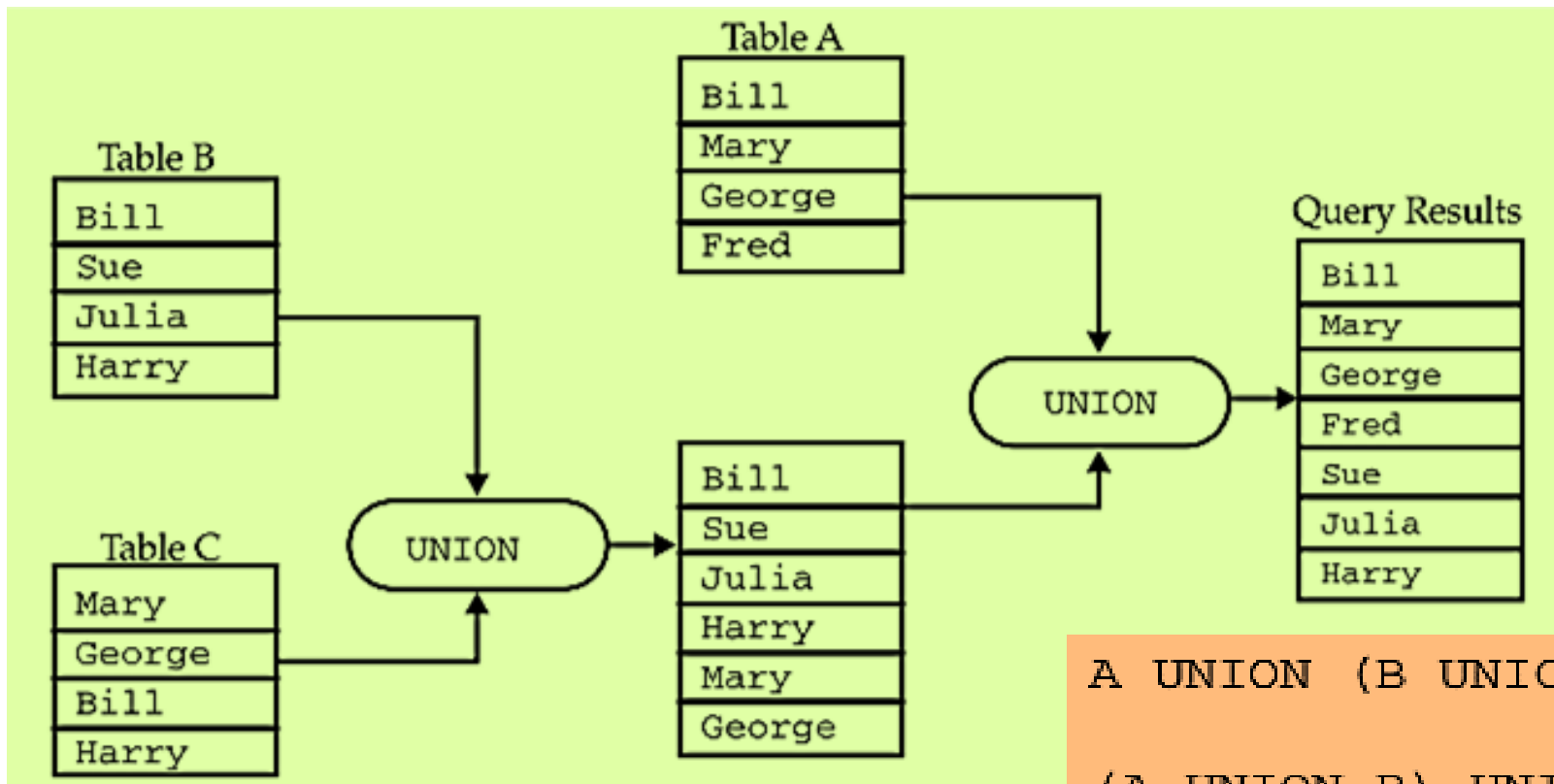
```
SELECT MFR_ID, PRODUCT_ID
FROM PRODUCTS
WHERE PRICE > 2000.00
UNION ALL
SELECT DISTINCT MFR, PRODUCT
FROM ORDERS
WHERE AMOUNT > 30000.00

ORDER BY 1, 2;
```

- List all the products where the price of product exceeds Rs.2,000 or where more than Rs. 30,000 of the product has been ordered in a single order, sorted by manufacturer and product number.

# Multiple UNIONS

- To combine three or more sets of query results



`A UNION (B UNION C)`

`(A UNION B) UNION C`

`(A UNION C) UNION B`