DSF-PT08_Phase 2

INTRODUCTION TO SQL

OBJECTIVES

- Understanding SQL
- Data-Types
- Writing table queries using SQL

SQL Overview

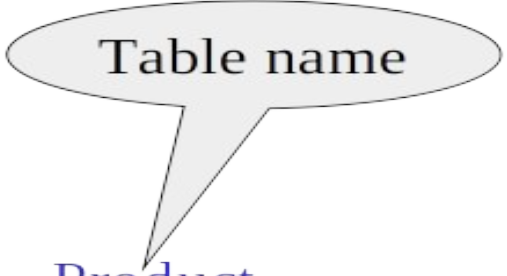
- Structured Query Language
- Developed in 1970.
- Helps control information stored in databases, allowing users to retrieve the specific information they are looking for.
- The standard for RDBMs

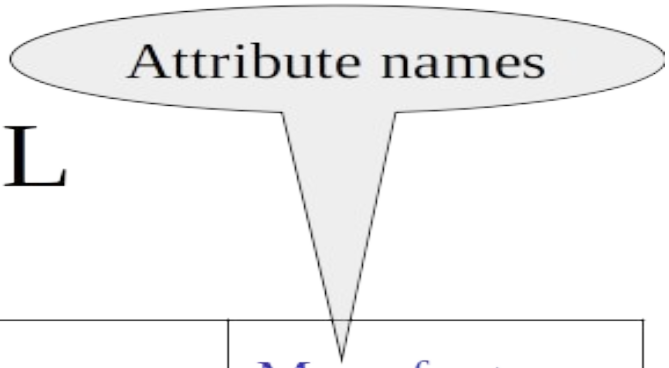
RDBMs

- A database management system that manages data as a collection of tables in which all relationships are represented by common values in related tables.

- Database Constraints:
  - Constraints are rules applied to columns or tables to enforce data integrity.
  - Examples include PRIMARY KEY, FOREIGN KEY, NOT NULL etc

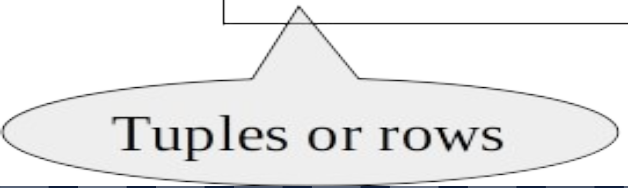
Tables in SQL



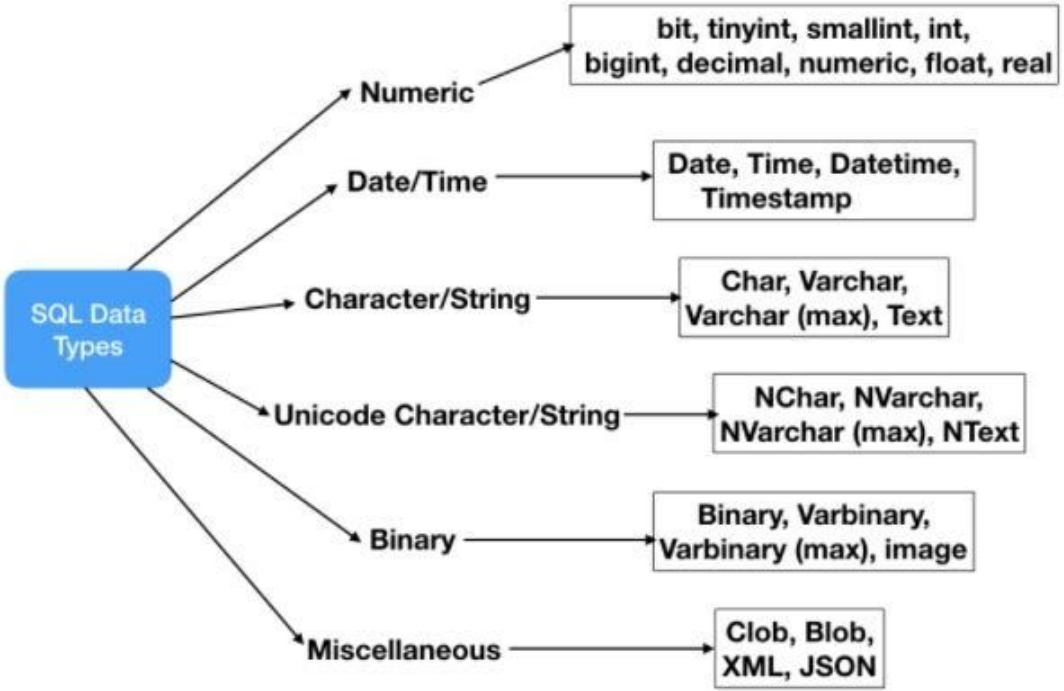


Product

PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

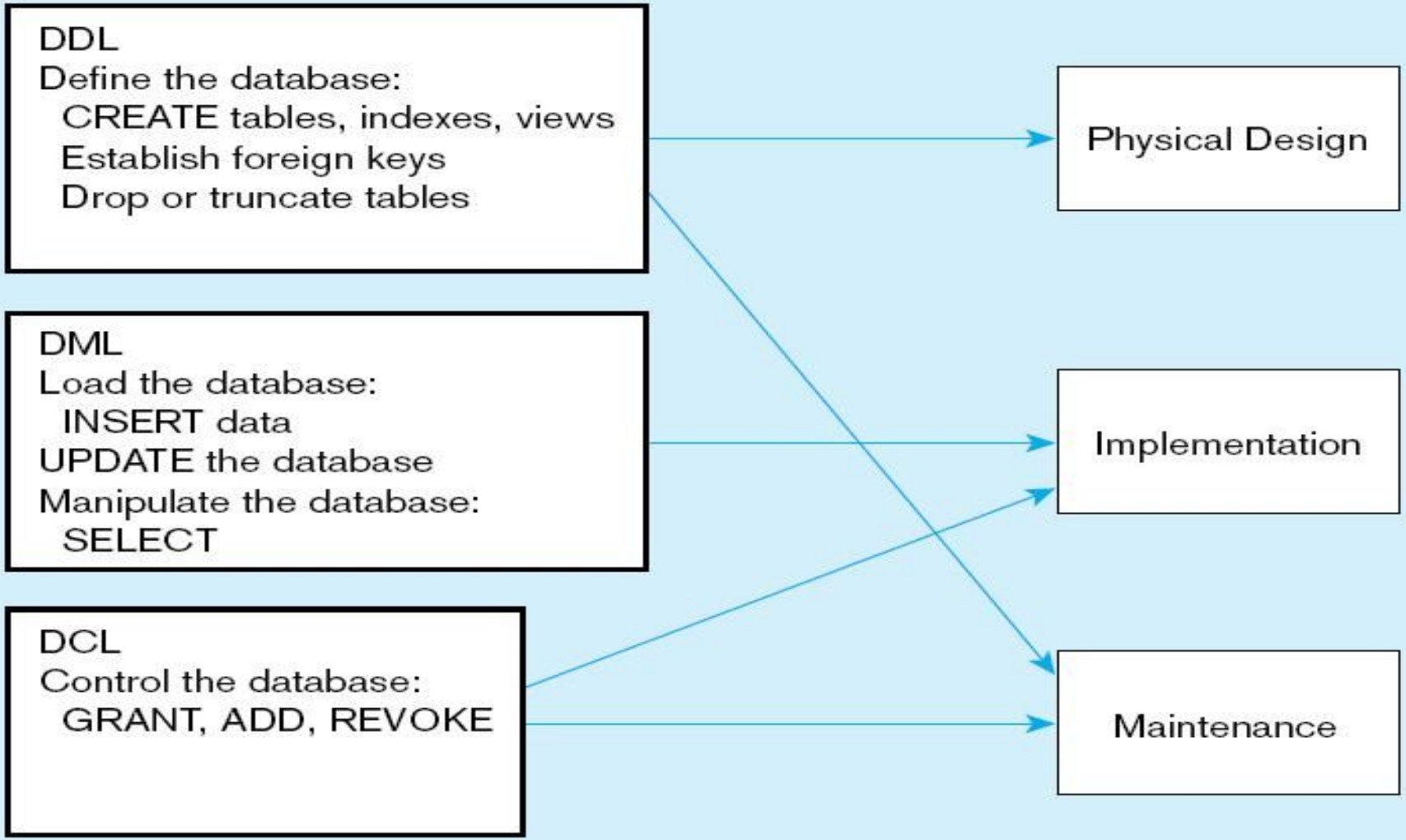


DATA TYPES IN SQL



SQL ENVIRONMENT

- Data Query Language (DQL)
  - Used for querying information from the database. SELECT
- Data Definition Language(DDL)
  - Commands that define a database, including creating, altering and dropping tables and establishing constraints. - CREATE, ALTER, DROP.
- Data Manipulation Language(DML)
  - Commands that maintain and query a database - INSERT, UPDATE, DELETE
- Data Control Language(DCL)
  - Commands that control a database, including administering privileges and committing data. - GRANT, REVOKE


DDL
Define the database:
CREATE tables, indexes, views
Establish foreign keys
Drop or truncate tables
DML
Load the database:
INSERT data
UPDATE the database
Manipulate the database:
SELECT
DCL
Control the database:
GRANT, ADD, REVOKE
Physical Design
Implementation
Maintenance

Data types in SQL

Numeric	INT or INTEGER: Integer data type (e.g., 1, 100, -5). SMALLINT: Small integer data type. BIGINT: Large integer data type. DECIMAL or NUMERIC: Fixed-point number with a specified precision and scale. FLOAT: Floating-point number. REAL: Single-precision floating-point number. DOUBLE PRECISION or FLOAT(n): Double-precision floating-point number.
String	CHAR(n): Fixed-length character string. VARCHAR(n): Variable-length character string with a maximum length of n. TEXT: Variable-length character string with no specified maximum length.
Boolean	BOOLEAN or BOOL: Represents true or false values.
Date and Time Types	DATE: Date (YYYY-MM-DD). TIME: Time of day (HH:MM:SS). DATETIME or TIMESTAMP: Combination of date and time. INTERVAL: Represents a period of time

Clauses in SQL

I. WHERE: Used to filter records based on a specified condition.
II. ORDER BY: Used to sort the result set in ascending or descending order.
III. GROUP BY: Used to group rows based on specified columns.
IV. HAVING: Used to filter the results of a GROUP BY clause.

Joins

Used to combine rows from two or more tables based on a related column.

- INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL OUTER JOIN.

Functions

SQL provides a variety of functions for data manipulation and analysis.

- COUNT, SUM, AVG, MAX, MIN.

LEFT JOIN

Everything on the left
+
anything on the right that matches

```
SELECT *
FROM TABLE_1
LEFT JOIN TABLE_2
ON TABLE_1.KEY = TABLE_2.KEY
```

ANTI LEFT JOIN

Everything on the left
that is NOT on the right

```
SELECT *
FROM TABLE_1
LEFT JOIN TABLE_2
ON TABLE_1.KEY = TABLE_2.KEY
WHERE TABLE_2.KEY IS NULL
```

RIGHT JOIN

Everything on the right
+
anything on the left that matches

```
SELECT *
FROM TABLE_1
RIGHT JOIN TABLE_2
ON TABLE_1.KEY = TABLE_2.KEY
```

ANTI RIGHT JOIN

Everything on the right
that is NOT on the left

```
SELECT *
FROM TABLE_1
RIGHT JOIN TABLE_2
ON TABLE_1.KEY = TABLE_2.KEY
WHERE TABLE_1.KEY IS NULL
```

OUTER JOIN

Everything on the right
+
Everything on the left

```
SELECT *
FROM TABLE_1
OUTER JOIN TABLE_2
ON TABLE_1.KEY = TABLE_2.KEY
```

ANTI OUTER JOIN

Everything on the left and right
that is unique to each side

```
SELECT *
FROM TABLE_1
OUTER JOIN TABLE_2
ON TABLE_1.KEY = TABLE_2.KEY
WHERE TABLE_1.KEY IS NULL
OR TABLE_2.KEY IS NULL
```

INNER JOIN

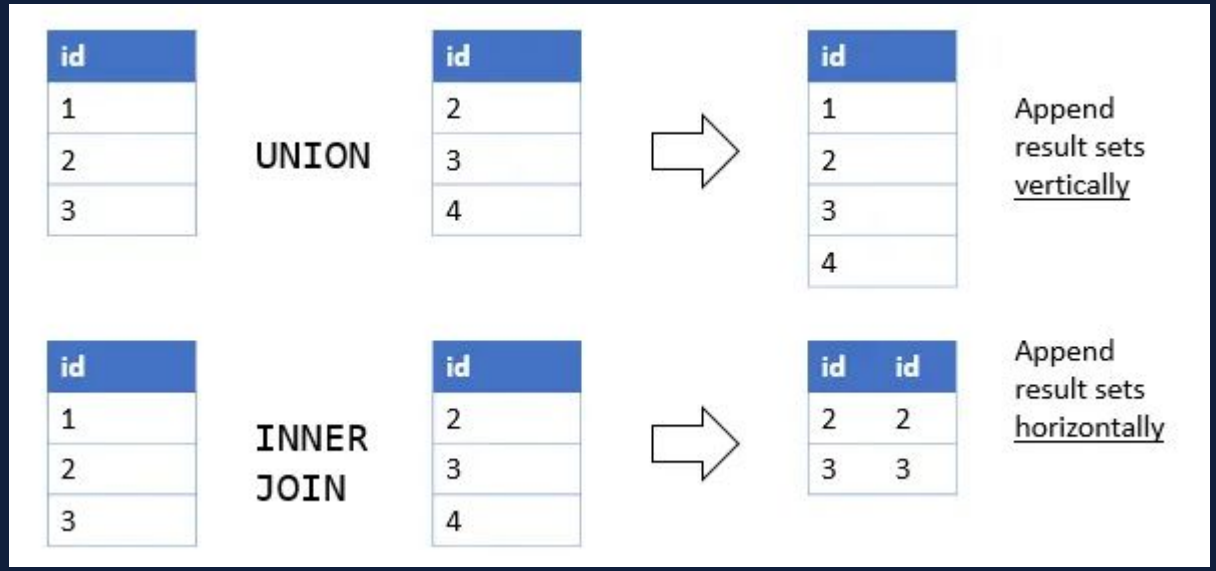
Only the things that match on the
left AND the right

```
SELECT *
FROM TABLE_1
INNER JOIN TABLE_2
ON TABLE_1.KEY = TABLE_2.KEY
```

CROSS JOIN

All combination of rows from the
right and the left (cartesean
product)

```
SELECT *
FROM TABLE_1
CROSS JOIN TABLE_2
```




```
SELECT *
FROM table1
UNION
SELECT *
FROM table2
```

NB: UNION can only be applied when

- The number of columns in both the table are same.
- The data types (integer / date / string) are same in the same order.