

MODULE 7 – SQL REVIEW

IT 207 – IT Programming



LECTURE OUTLINE

- ❖ Overview of Databases
- ❖ Overview of Relational Database Concepts
- ❖ Overview of SQL

WHAT IS A DATABASE ?

- ❖ Databases can be thought of as an organized collection of information.
- ❖ Data refers to individual, raw facts.
- ❖ Information refers to the insights and inferences that results from processing information
- ❖ A a collection of data files that holds information pertaining to a specific field/activity/area/context over a long period of time
- ❖ A very pervasive and prevalent data storage system that underlies a lot of daily used applications

WHAT IS A DBMS?

- ❖ A DBMS is an electronic record-keeping system for fast and convenient access to the information held in a Database
- ❖ DBMS provides...

efficient, reliable, convenient, and safe multiuser storage of and access to massive amounts of persistent data

DATABASE VS. DATABASE MANAGEMENT SYSTEM

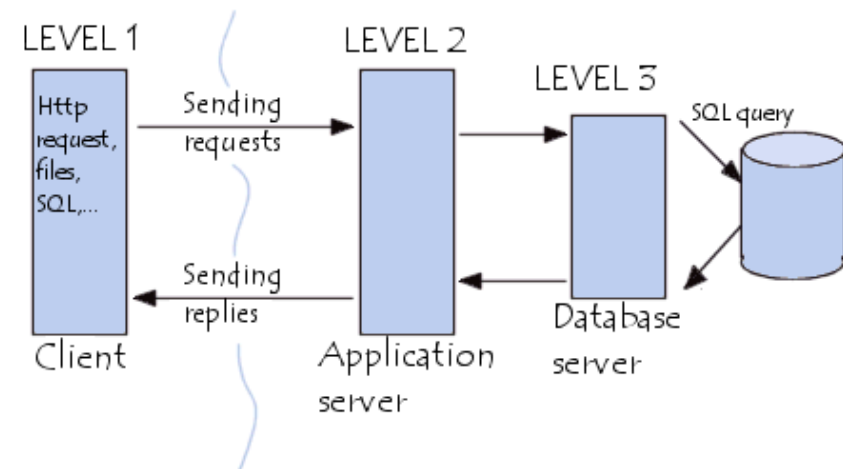
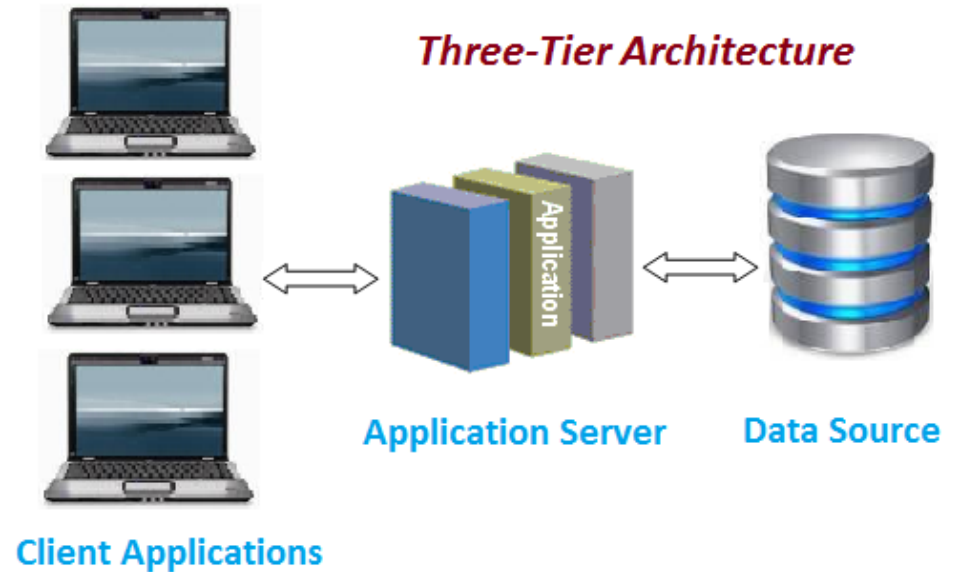
- ❖ A database is
 - ❖ a representation of data that can be read from and written to and is often stored separately from any application that uses the data.
 - ❖ made available to multiple applications.
- ❖ A database management system (DBMS)
 - ❖ is a software system that manages databases.
 - ❖ executes commands, provides security, enables network access, and provides admin tools for database administrators (DBAs) to work with database files.

EXAMPLES

- ❖ Search engines, banking systems, airline reservations, corporate records, payrolls, sales inventories.
- ❖ New applications: Wikis, social/biological/multimedia/scientific/geographic data, heterogeneous data.
- ❖ On cell phones: contact list, calendar, email apps, photo galleries, social networking apps, and shopping apps.

3 TIER ARCHITECTURE

- ❖ A 3 Tier Architecture is the most popular client server architecture in DBMS
- ❖ A 3-tier architecture has the following layers:
 - ❖ Presentation layer (your PC, Tablet, Mobile, etc.)
 - ❖ Application layer (server)
 - ❖ Database Server



TYPES OF DATABASES

❖ Structured:

- ❖ Consists of clearly defined data types with patterns that make them easily searchable
- ❖ A **Relational** databases is a typical example of a structured database

❖ Unstructured:

- ❖ “everything else”—is composed of data that is usually not as easily searchable, including formats like audio, video, and social media postings.

[Structured vs. Unstructured Data: Key Differences Explained](#)

RELATIONAL DATABASE SYSTEM

- ❖ The most mature database systems in production
 - ❖ Based on the relational model
 - ❖ Considered the norm in DBMS.
 - ❖ From a user perspective: data is represented as a set of records (rows) grouped in relations (tables)
 - ❖ From an implementation perspective: implemented as a complex data structure (B+ trees) that allow rapid response to a variety of queries.
 - ❖ Managed by Structured Query Language (SQL)

RELATIONAL DATABASE CONCEPTS



DATABASE SCHEMA

- ❖ Schema is the logical structure of data
- ❖ In relational databases (RDB) the schema specifies the following:
 - ❖ The database **entities** and their names
 - ❖ Entity is the name given to any piece of data.
 - ❖ The **attributes** for each entity and the data type of each attribute.
 - ❖ The **constraints** that need to be applied to attribute.
 - ❖ **Entity integrity**: guarantees that one or more **attributes solely identifies** each entity within the database.
 - ❖ **Referential integrity**: guarantees that if an **attribute is used to refer** to an entity within a relation then the referenced instance **must exist**.

SCHEMA REPRESENTATION IN RDB

- ❖ Entities are represented as tables or relations
- ❖ Attributes of an entity are represented by the columns of the table.
- ❖ Entity integrity constraint within an entity table is applied by defining a Primary key.
 - ❖ A Primary key: a column (or group of columns) in a relational database table that's distinctive for each row (record). It uniquely identifies each row.
- ❖ Referential integrity constraint within an entity table is applied by defining Foreign Keys.
 - ❖ A Foreign Key: a columns (or collection of columns) in one table, that refers to the Primary Key(s) in another table. It is used to establish and enforce a link between the data stored in two tables
 - ❖ Any value used as a foreign key must exist as a primary key in the related table.

SCHEMA EXAMPLE

Entity Name

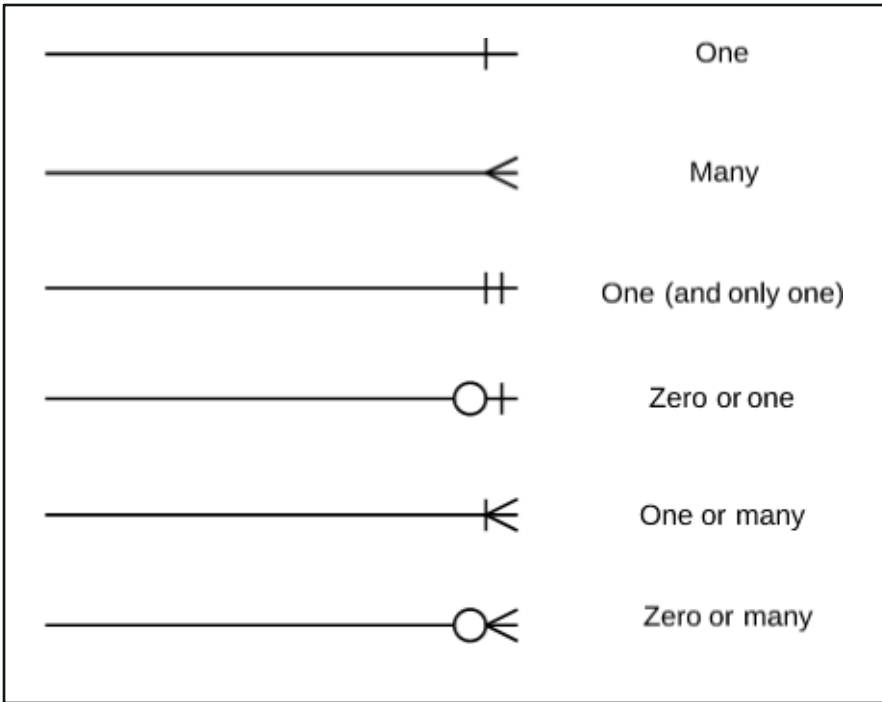
Student	
PK	<u>StudentID INT</u>
	FirstName VARCHAR(25)
	LastName VARCHAR(50)

Underline Primark Key attribute

Links between tables are identified using foreign keys

Enrollment	
PK,FK1	<u>StudentID INT</u>
PK,FK2	<u>ClassID INT</u>

Schema Connectors



Class	
PK	<u>ClassID INT</u>
	Course VARCHAR(25)
	Schedule VARCHAR(10)
	Room VARCHAR(6)
	Team VARCHAR(6)

Attribute & their data types

Attributes written in bold must have a value (Cannot be Null)

SCHEMA INSTANTIATION EXAMPLE

Student	
PK	<u>StudentID INT</u>
	FirstName VARCHAR(25)
	LastName VARCHAR(50)

Enrollment	
PK,FK1	<u>StudentID INT</u>
PK,FK2	<u>ClassID INT</u>

Class	
PK	<u>ClassID INT</u>
	Course VARCHAR(25)
	Schedule VARCHAR(10)
	Room VARCHAR(6)
	Team VARCHAR(6)

Student

StudentID	FirstName	LastName
123	Anne	Gary
456	Bob	Dany
789	Tracy	Blond



Each row is called a 'tuple' and represents an instance of the schema

Enrollment

StudentID	ClassID
123	106
123	214
456	206
456	207
789	109
789	214

Class

ClassID	Course	Schedule	Room	Team
106	Java Prog			
109	Python Prog			
214	DB Fund			
206	OOP			
207	IT Prog			

DATABASE LANGUAGES



DATABASE LANGUAGES

- ❖ Database languages are **high level languages** that are used to interact with the database.
- ❖ **Data Definition Language (DDL)**
 - ❖ Sets up the schema for the database
- ❖ **Data Manipulation Language (DML)**
 - ❖ Queries and modifies the data in the database

STRUCTURED QUERY LANGUAGE (SQL)

- ❖ Structured Query Language (SQL) is the principal language used to describe and manipulate relational databases.
- ❖ Data definition part of SQL is used for defining schemas (DDL)
- ❖ Data manipulation part of SQL is used for querying and modifying the database (DML)
- ❖ SQL is a declarative language (C++/Java are imperative languages)
 - ❖ Say “what to do” rather than “how to do it”.
 - ❖ The DBMS figures the “best” way to execute SQL statements (considers query optimization).

SQL TERMINOLOGY

- ❖ Statement: refers to any complete SQL command.
- ❖ Query: technically refers to a statement that retrieves data from a database
 - ❖ In DB jargon is equivalent to statement
- ❖ Clause: refers to a a section of a query.

All SQL statements must end with a **semicolon (;)**

SQL DATA DEFINITION LANGUAGE



SQL DDL

- ❖ **CREATE TABLE:** creates a relation and declares its schema

```
CREATE TABLE <table name> ( <list of attribute names separated by comma> );
```

- ❖ Each attribute name will be a column
- ❖ The data type of each attribute must be defined

- ❖ **ALTER TABLE:** modifies the schema of a table

- ❖ **ADD** adds a new attribute(s) to the schema

```
ALTER TABLE <Table name> ADD <attribute-name attribute type> ;
```

- ❖ **DROP** deletes an attribute(s) from the schema

```
ALTER TABLE <Table name> DROP <attribute-name >;
```

- ❖ **DROP TABLE:** deletes the table

```
DROP TABLE <table name> ;
```

SQL DATA TYPES

❖ SQL provide several data types for the following

- ❖ Integer types
- ❖ Decimal
- ❖ String types
- ❖ Date/time types

❖ Most used data types:

- ❖ INT
- ❖ DECIMAL
- ❖ VARCHAR
- ❖ DATE
- ❖ TIME

DEFAULT VALUES

- ❖ Attributes in relational databases need to hold an initial values.
- ❖ **NULL** is a predefined value indicating that the value of an attribute is missing or unknown.
 - ❖ NULL is different from zero
- ❖ The **DEFAULT** keyword could be used to indicate an appropriate value for an attribute if no value was initially given.

KEY ATTRIBUTES

- ❖ A key attribute (set of attributes) is an attribute (a set) that can identify the tuple (an ordered sequence of values)
 - ❖ Duplicates are not allowed for keys
- ❖ **PRIMARY KEY** or **UNIQUE** are two keywords used to specify the key attribute(s)
 - ❖ Only one PRIMARY KEY is allowed for a relation, but there could be several UNIQUE keys.
 - ❖ An attribute declared as PRIMARY KEY is not allowed to take on NULL as an initial value.
 - ❖ An attribute declared as UNIQUE can take on NULL as an initial value.

DECLARING KEYS

❖ Keys are defined using two methods

❖ Method 1:

- ❖ suitable when the key is a single attribute
- ❖ Specify adding the PRIMARY KEY/UNIQUE keyword to the attribute declaration within the schema

❖ Method 2:

- ❖ Suitable when the key is a single attribute or multiple attributes
- ❖ A list of the key attribute(s) is/are specified in the schema

```
CREATE TABLE STUDENT(  
    StudentID INT PRIMARY KEY,  
    FirstName VARCHAR(25),  
    LastName VARCHAR(50),  
);
```

```
CREATE TABLE STUDENT(  
    StudentID INT,  
    FirstName VARCHAR(25),
```

```
CREATE TABLE ENROLLMENT(  
    StudentID INT,  
    ClassID INT,  
    PRIMARY KEY (StudentID, ClassID ),  
    :::  
/*Constraints in the coming slides*/  
);
```


ADDING FOREIGN KEYS

- ❖ A FOREIGN KEY:
 - ❖ defines the link between two or more tables
 - ❖ has the primary key for one of the related tables.
- ❖ A FOREIGN KEY is defined using the **CONSTRAINT** and the **REFERENCES** keywords
- ❖ Referential Integrity is maintained using the **CASCADE** option
 - ❖ **CASCADE** propagates any changes happening to an existing **PRIMARY KEY** value to the corresponding **FOREIGN KEY**

```
CREATE TABLE ENROLLMENT(  
    StudentID INT,  
    ClassID INT,  
    PRIMARY KEY (StudentID, ClassID ),  
    CONSTRAINT FOREIGN KEY (StudentID)  
    REFERENCES Student (StudentID)  
    ON UPDATE CASCADE ON DELETE CASCADE,  
    CONSTRAINT FOREIGN KEY (ClassID)  
    REFERENCES Course (ClassID)  
    ON UPDATE CASCADE ON DELETE CASCADE);  
);
```

SQL DATA MANIPULATION LANGUAGE



DATA MODIFICATION

- ❖ Insert tuples into the relation R

```
INSERT INTO R (a1, ... an) VALUES (v1, ..., vn);
```

- ❖ Delete tuples from relations

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

- ❖ Update existing tuples

```
UPDATE R SET < new-value assignments> WHERE <CONDITION>;
```

THE SELECT STATEMENT

- ❖ SELECT is the most common statement used in SQL
- ❖ SELECT is used to retrieve data from one or more tables in a database.
- ❖ SELECT has a rigid syntax
 - ❖ SELECT: specifies the column names
 - ❖ FROM: specifies the table name
 - ❖ WHERE: specifies a criteria (optional)
 - ❖ ORDER: specifies ordering of results (Optional)
- ❖ SELECT * selects all fields (columns) in the table

```
SELECT field1, field2, field3  
FROM table1  
WHERE criteria  
ORDER BY field1, field2;
```

SUMMARY

- ❖ Database is a persistent storage for collection of data files managed by Database management systems
- ❖ A 3 Tier Architecture is the most popular client server architecture in DBMS
- ❖ The relational Databases is the most mature and popular databases used in production
- ❖ DB schema defines the logical structure of the data in the database
- ❖ Relational databases represent entities as tables and attributes as columns
- ❖ In RDB entity integrity and referential integrity are maintained by using primary keys and foreign keys respectively
- ❖ Structured Query Language (SQL) is the principal language used to describe and manipulate relational databases.
- ❖ The SELELCT statement is the most common statement used in SQL to query the database.