Introduction to RDBMS & SQL

View of Data

- One main purpose of a DBMS:
  - Provide users with abstract view of data
    - Hiding details of how data are stored and maintained
    - Hiding the complexity from the naïve users

Consider a C program:

```
#include<stdio.h>
int main(){
int a;
printf("Enter a number");
scanf("%d",&a);
printf("The entered number is %d", a);
return 0;
}
```

- The view for the users is the executed printf and scanf options.
- The program has a logic behind this with a set of statements.
- The physical aspect of this program is with regard to its storage space requirements.

Data Models

- Data model:
  - A collection of conceptual tools for describing data, data relationships, data semantics, and consistency constraints.
- The data models can be classified into four different categories:
  - Relational Model
  - Entity-Relationship Model
  - Semi-structured Data Model
  - Object-Based Data Model

Relational Model

- In the relational model, data are represented in the form of tables.
- Each table has multiple columns, and each column has a unique name.
- Each row of the table represents one piece of information.

A sample relational database

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

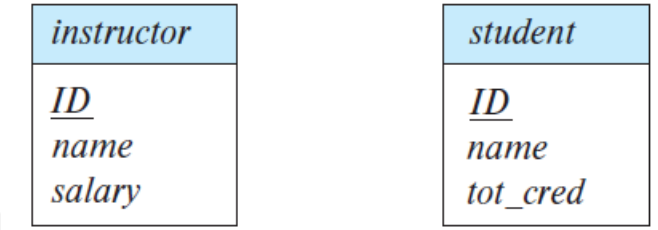
(a) The *instructor* table

dept_name	*building*	*budget*
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

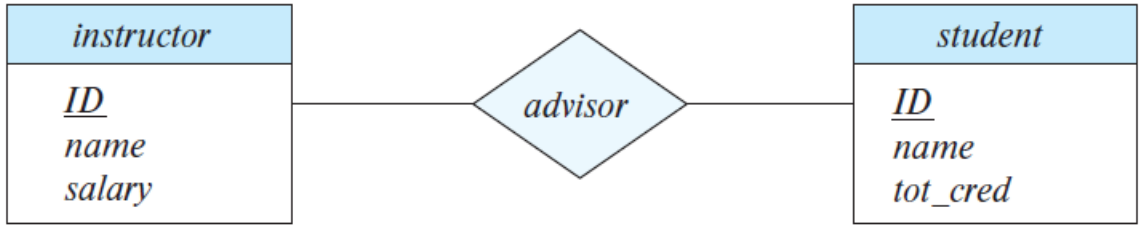
(b) The *department* table

Entity-Relationship Model

- Diagrammatic representation.
- Collection of objects called Entities.
- Relationship between the entities.
- Commonly used in database design.
- Example:
  - Entity - instructor and Student
  - Relationship - advisor



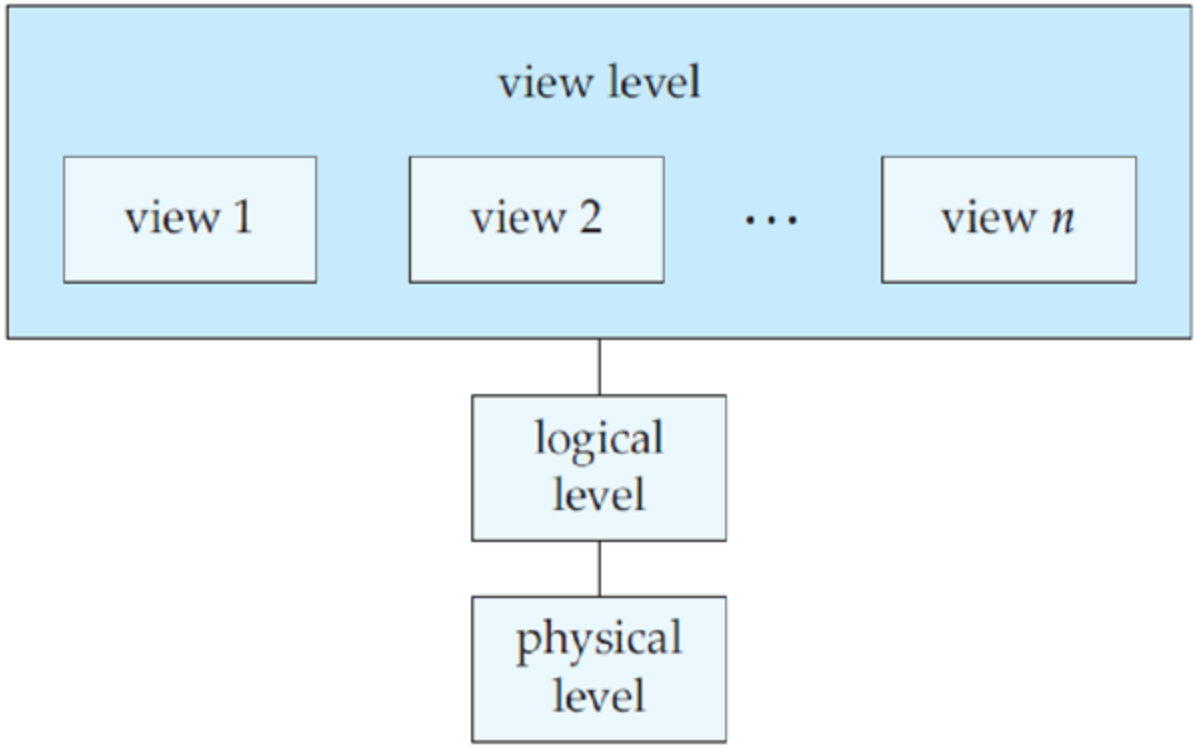
E-R diagram showing entity sets *instructor* and *student*.



E-R diagram showing relationship set *advisor*.

Three level Data Abstraction

- ANSI/SPARC Architecture
- The three levels are:
  - Physical level
  - Logical level
  - View level



Three Levels of Data Abstraction

- Physical Level
  - The lowest level of abstraction describes *how* the data are actually stored.
  - The physical level describes complex low-level data structures in detail.
- Logical Level
  - Describes *what* data are stored in the database, and what relationships exist among those data.
  - Therefore, the entire database in terms of a small number of relatively simple structures.
- View Level
  - The highest level of abstraction describes only part of the entire database.
  - Simplifies the interaction between user and the system.

Instances And Schema

- The collection of information stored in the database at a particular moment is called an instance of the database.
- The overall design of the database is called the database schema.
- Based on the levels of abstraction:
  - Physical Schema – Database design at the physical level
  - Logical schema – Database design at the logical level
  - Subschemas – Different views of the database

Example of schemas

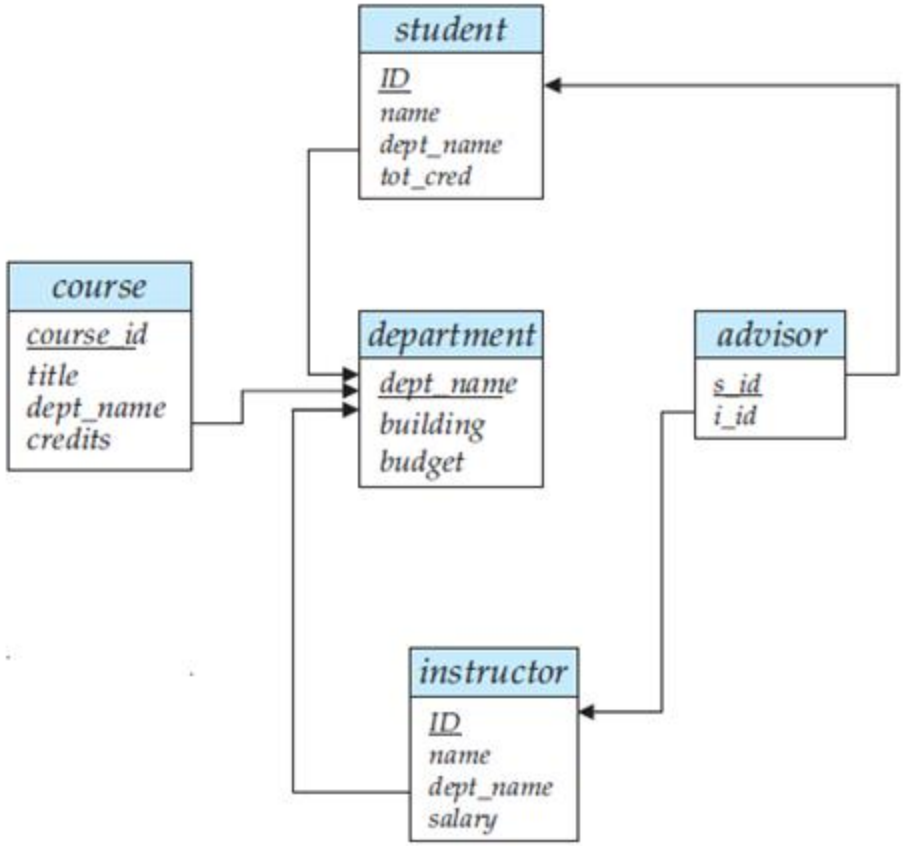
Department (dept_name, building, budget)

Student (*ID*, *name*, *dept name*, *tot cred*)

Instructor (id, name, dept_name, salary)

Advise (*s id*, *i id*)

Course (Course_id, title, dept_name, credits)



Example of Instance

- Information stored in the database at a particular moment
- E.g. Set of instructor's information at a particular time *t*
- Insertions, Updates, Deletions to this data later, *t+1*
  - new instance

ID	*name*	*salary*	*dept_name*	*building*	*budget*
22222	Einstein	95000	Physics	Watson	70000
12121	Wu	90000	Finance	Painter	120000
32343	El Said	60000	History	Painter	50000
45565	Katz	75000	Comp. Sci.	Taylor	100000
98345	Kim	80000	Elec. Eng.	Taylor	85000
76766	Crick	72000	Biology	Watson	90000
10101	Srinivasan	65000	Comp. Sci.	Taylor	100000
58583	Califieri	62000	History	Painter	50000
83821	Brandt	92000	Comp. Sci.	Taylor	100000
15151	Mozart	40000	Music	Packard	80000
33456	Gold	87000	Physics	Watson	70000
76543	Singh	80000	Finance	Painter	120000

Database Constraints

- Constraints are certain rules defined to limit the type of data inserted into the table.
- It ensures the reliability and accuracy of data in the database.
- It can be applied either at the column level or at the table level.
- Defined either during the table creation or later with some database commands.

Most Commonly used Constraints

- Unique Constraint
- Not null Constraint
- Default Constraint
- Check Constraint

Unique Constraint

- It's a column level constraint.
- Ensures that all values in a column are different.
- It's different than *primary key*.

Not Null Constraint

- It's a column level constraint.
- Ensures that a column cannot have a NULL value.

Default Constraint

- It's a column level constraint.
- Provides a default value for a column where no value specified during insertion of record.

Check Constraint

- It's a column level constraint.
- Ensures that all the values in a column satisfy certain conditions.

Database Keys & Constraints

You can define this column with Check Constraint

Primary Key

Unique Key

Table: Students

Roll No	Name	Address	Date of Birth	Registration No.
BCA101	Alan	Mumbai	20-10-1989	6087452099
BCA102	Dev	Chennai	05-02-1990	9789112902
BCA103	Ziva	Kolkata	27-05-1989	8340008479

You can define this column with Not NULL Constraint

You can define this column with Default Constraint

Foreign Key

Table: Marks

Roll No	Semester	Subject	Marks	Faculty
BCA102	II	Maths	89	Helen
BCA103	I	English	94	Yami
BCA101	IV	C/C++	90	Ann
BCA103	II	Maths	91	Helen
BCA101	III	Computer	95	John

Need of Database Keys

- In today's world, the data is rapidly increasing, GB to TB of data generating per day.
- Challenge: Instantly store and retrieve the large set of data in real-time.
- Database Keys: Ensures each record in a table is precisely identified.

Database Keys

- Keys are fields in a database table which participate in below activities in RDBMS systems:
  - Establish relationships between multiple tables in the database.
  - To maintain uniqueness in a table.
  - To keep consistent and valid data in database.
  - Might help in fast data retrieval by facilitating indexes on column(s).

"A Key is either a single column (or an attribute) or a group of columns that can uniquely identify rows (or tuples) in a table."

Types of Keys

- Candidate Key
- Primary Key
- Unique Key
- Super Key
- Foreign Key

Candidate Key

- The minimal set of fields which can uniquely identify each record in a table.
- It is an attribute or a set of attributes that can act as a Primary Key for a table to uniquely identify each record in that table.
- There can be more than one candidate key.

Candidate Keys

Table: Employee

Employee Table				
Employee_ID	Employee_Name	Address	**License_Number**	**Passport_Number**
Emp101	John	Mumbai	MI490587	WBH874520995
Emp102	Riya	New Delhi	DL123456	SQV789112902
Emp103	Bob	Chennai	CH231458	EHN340008479

Criteria for a Candidate Key

- It must contain unique values.
- It cannot contain null values.
- It comprises a minimum number of fields necessary to define uniqueness.
- Its values must uniquely and exclusively identify each record in the table.
- Its value must exclusively identify the value of each field within a given record.
- Its value can be modified only in rare or extreme cases.

Primary Key

- A primary key field identifies the table throughout the database structure and helps establish relationships with other tables.
- It's a candidate key which is most appropriate to become main key of any table.
- A primary key value uniquely identifies a given record within a table and represents that record throughout the entire database. It also helps to guard against duplicate records.

Table: Employee

Primary Key

Employee Table				
Employee_ID	Employee_Name	Address	License_Number	Passport_Number
Emp101	John	Mumbai	MI490587	WBH874520995
Emp102	Riya	New Delhi	DL123456	SQV789112902
Emp103	Bob	Chennai	CH231458	EHN340008479

Criteria for a Primary Key

- It must contain unique values.
- It cannot contain null values.
- It comprises a minimum number of fields necessary to define uniqueness.
- Its values must uniquely and exclusively identify each record in the table.
- Its value must exclusively identify the value of each field within a given record.
- Its value can be modified only in rare or extreme cases.

Unique Key

- The set of one or more attribute that can be used to uniquely identify the records in table.
- Unique key is similar to primary key **but** unique key field can contain a "**Null**" value but primary key doesn't allow "**Null**" value.

Unique Keys

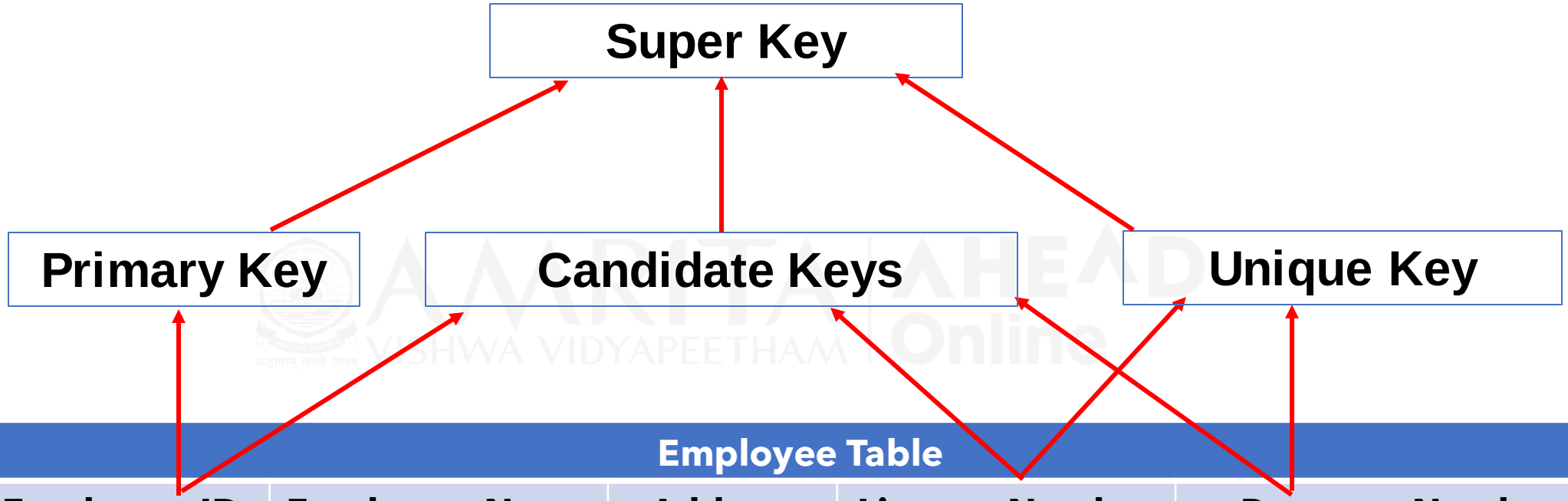
Table: Employee

Employee Table				
Employee_ID	Employee_Name	Address	**License_Number**	**Passport_Number**
Emp101	John	Mumbai	MI490587	WBH874520995
Emp102	Riya	New Delhi	DL123456	SQV789112902
Emp103	Bob	Chennai	CH231458	EHN340008479

Super Key

- A set of one or more than one keys that can be used to uniquely identify the record in table.
- It is a set of one or more attributes whose combined value uniquely identifies the entity in the entity set.
- It is simply a non-minimal Candidate Key, that is to say one with additional columns not strictly required to ensure uniqueness of the row.

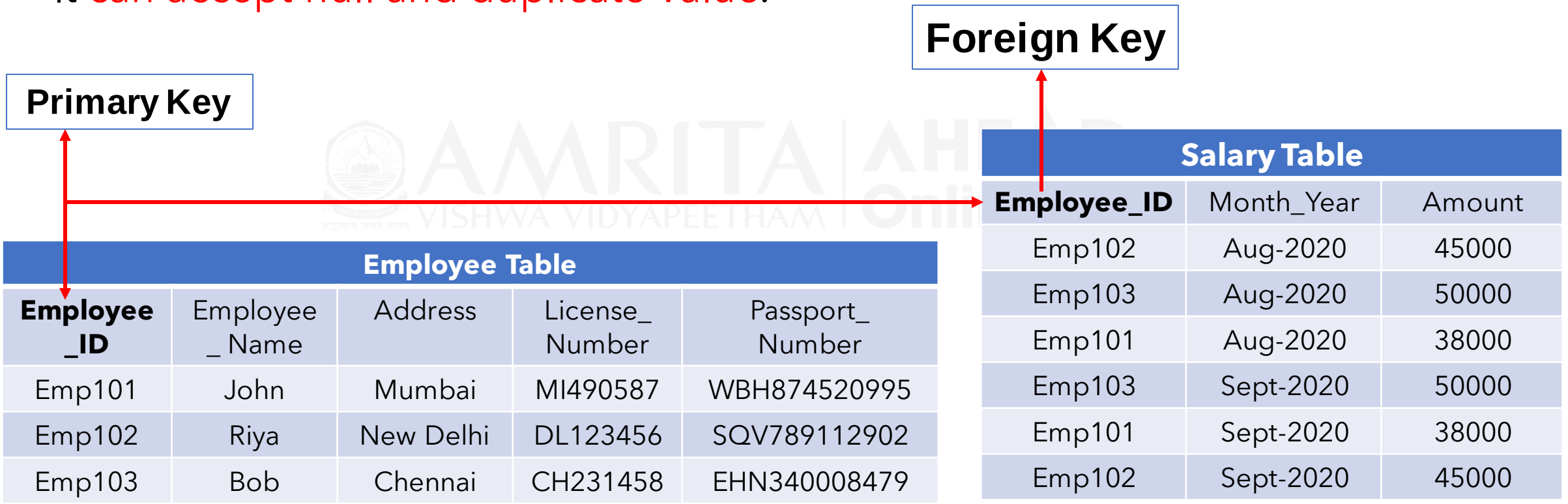
Super Key



Employee Table				
Employee_ID	**Employee_Name**	**Address**	**License_Number**	**Passport_Number**
Emp101	John	Mumbai	MI490587	WBH874520995
Emp102	Riya	New Delhi	DL123456	SQV789112902
Emp103	Bob	Chennai	CH231458	EHN340008479

Foreign Key

- It is used to generate the relationship between the tables.
- It is a field in database table that is Primary key in another table.
- It can accept null and duplicate value.



Employee Table

Employee_ID	Employee_ Name	Address	License_ Number	Passport_ Number
Emp101	John	Mumbai	MI490587	WBH874520995
Emp102	Riya	New Delhi	DL123456	SQV789112902
Emp103	Bob	Chennai	CH231458	EHN340008479

Salary Table

Employee_ID	Month_Year	Amount
Emp102	Aug-2020	45000
Emp103	Aug-2020	50000
Emp101	Aug-2020	38000
Emp103	Sept-2020	50000
Emp101	Sept-2020	38000
Emp102	Sept-2020	45000

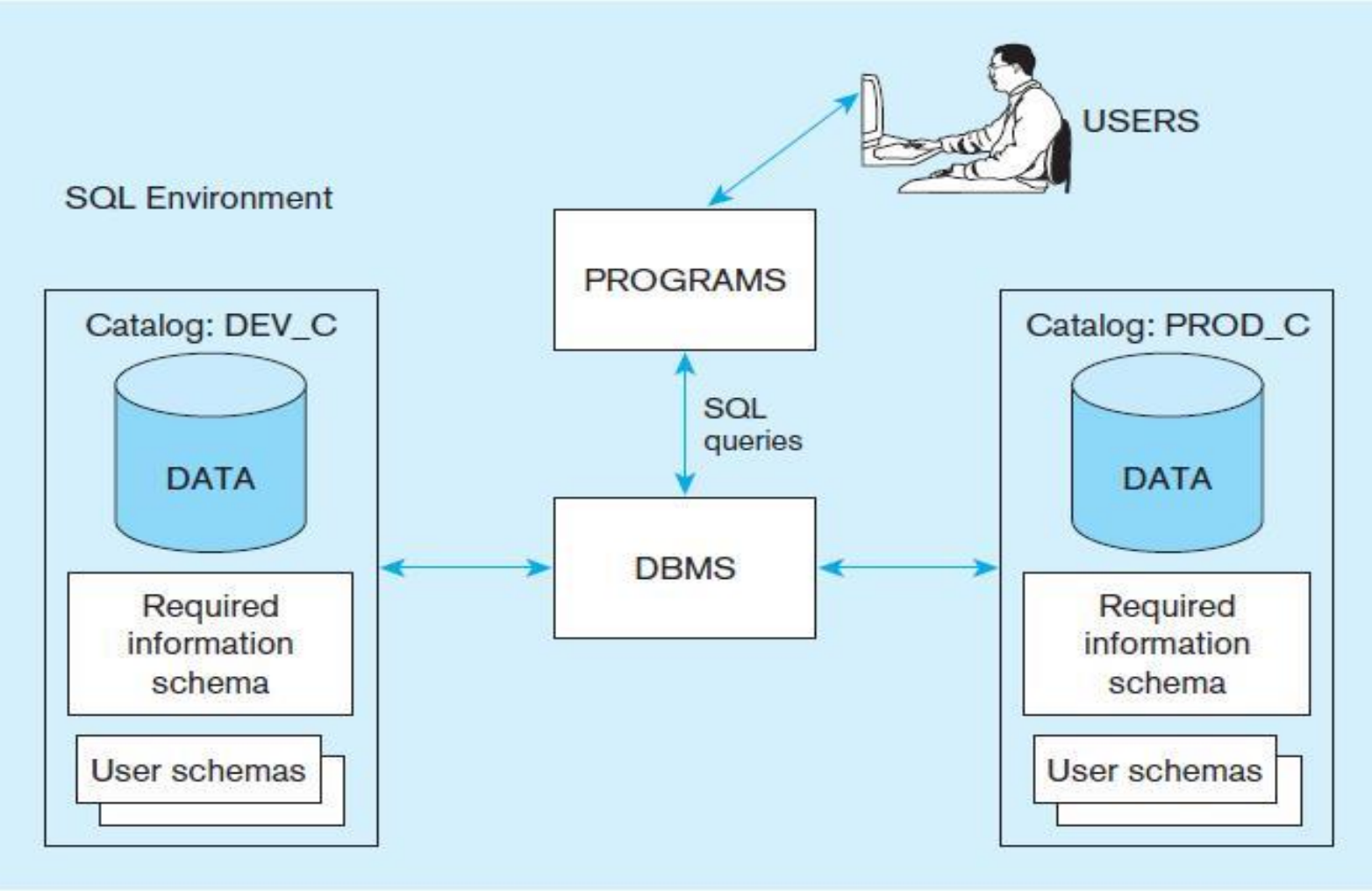
SQL

- SQL - Structured Query Language : A relational database language.
- It is a standard programming language designed for Storing, Retrieving and Manipulating the data that exist inside a RDBMS.
- It is a medium used to communicate to the DBMS.
- SQL commands consist of English like statements which are used to query, insert, update and delete data.
- It is also referred to as "*non-procedural database language*".

Features of SQL

- SQL is a language used to interact with the database.
- SQL is a data access language.
- SQL is based on relational tuple calculus.
- SQL is a standard Relational Database Management Language.
- The first commercial DBMS that supported SQL was Oracle in 1979.
- SQL is a "nonprocedural" or "declarative" language.

The SQL Environment



SQL Datatypes

	Data Type	Description
String	CHARACTER (CHAR)	Stores string values containing any characters in a character set. CHAR is defined to be a fixed length.
	CHARACTER VARYING (VARCHAR or VARCHAR2)	Stores string values containing any characters in a character set but of definable variable length.
	BINARY LARGE OBJECT (BLOB)	Stores binary string values in hexadecimal format. BLOB is defined to be a variable length. (Oracle also has CLOB and NCLOB, as well as BFILE for storing unstructured data outside the database.)
Number	NUMERIC	Stores exact numbers with a defined precision and scale.
	INTEGER (INT)	Stores exact numbers with a predefined precision and scale of zero.
Temporal	TIMESTAMP TIMESTAMP WITH LOCAL TIME ZONE	Stores a moment an event occurs, using a definable fraction-of-a-second precision.Value adjusted to the user's session time zone (available in Oracle and MySQL)
Boolean	BOOLEAN	Stores truth values: TRUE, FALSE, or UNKNOWN.

What can SQL do ?

- Execute queries against a database.
- Retrieve data from a database.
- Insert, Update, Delete records in a database.
- Create new databases.
- Create new tables, stored procedures, views in a database.
- Set permissions on tables, procedures, and views.

Types of SQL Commands

- DDL - Data Definition Language
- DML – Data Manipulation Language
- DCL – Data Control Language
- TCL - Transaction Control Language

DDL - Data Definition Language

- DDL commands can be used to define the database schema.
- DDL commands are :
  - CREATE – Create an object. i.e. create a database, table, triggers, index, functions, stored procedures etc.
  - DROP - To delete objects. Eg: delete tables, delete a database, etc.
  - ALTER - Used to alter the existing database or its object structures.
  - TRUNCATE - Removes records from tables.
  - RENAME - Renaming the database objects.

DML - Data Manipulation Language

- DML commands are used for performing queries on the data within schema objects.
- DML commands are :
  - SELECT – Select records or data from a table.
  - INSERT INTO – Insert data into a database table.
  - UPDATE – Update existing records within a table.
  - DELETE – Delete unwanted records from a table.

DCL - Data Control Language

- DCL commands will control the Data access permission.
- DCL commands are :
  - GRANT – It permits users to access the database.
  - REVOKE – It withdraws the permission given by GRANT to access the database.

TCL - Transaction Control Language

- TCL commands deals with the transaction within the database.
- TCL commands are :
  - COMMIT – Commit the running transaction.
  - ROLLBACK – Rollback the current transaction.
  - SAVEPOINT – Set a save point so that, next time it will start from here.
  - SET TRANSACTION – Specify the characteristics of the transactions.

Summary

- In this session, we discussed about an overview of Structured query language (SQL).
- Evolution and benefits of SQL.
- A simplified schematic of a typical SQL Environment.
- Datatypes in SQL.

Reference & Image Courtesy

- Sumathi, Sai, and S. Esakkirajan. *Fundamentals of relational database management systems*. Vol. 47. Springer, 2007.
- Modern Database Management, 10th Edition By Jeffrey A. Hoffer, 2011.