Business Databases

Lecture 1 - Introduction to database management

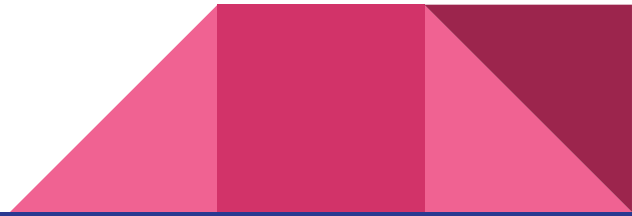
Ms. Himani Dudhat
Product Analyst, Google Pay

Agenda

- Course introduction
- What is a database?
- Relational database
- ER model for relational database
- Database keys
- What is SQL?
- Basics of table

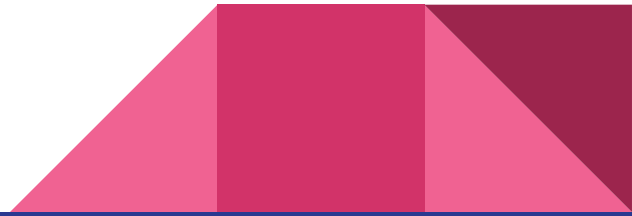
Course Introduction

- 2 credits subject
- **Goal** : In-depth understanding of concepts for retrieving and transforming data from a database
- Industry importance

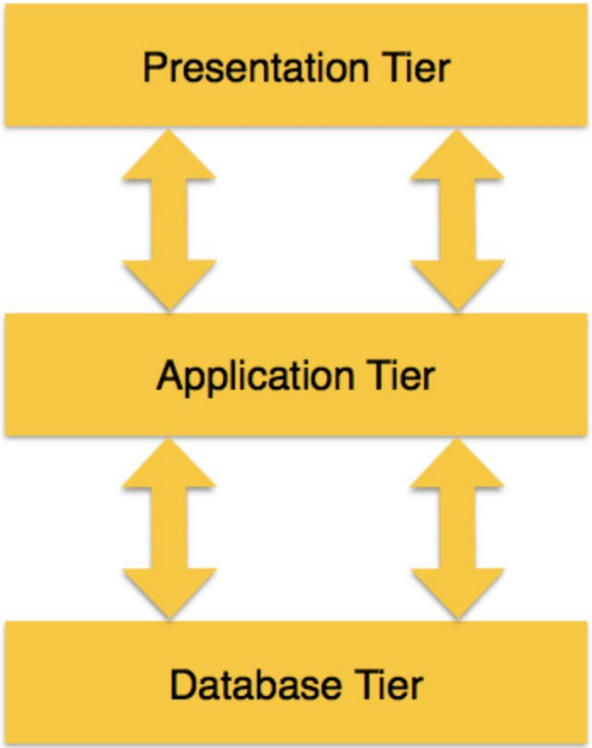


What is a database?

- Database is a collection of related data and data is a collection of facts and figures that can be processed to produce information.
- A database management system (DBMS) stores data in such a way that it becomes easier to retrieve, manipulate, and produce information.
- Why understand DBMS architecture?



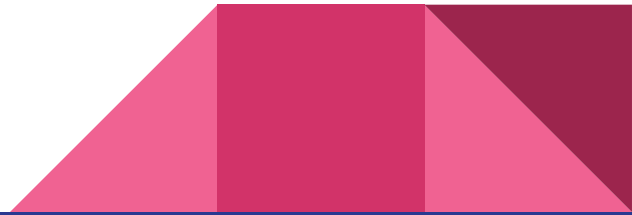
3-tier Architecture



- A 3-tier architecture separates its tiers from each other based on the complexity of the users and how they use the data present in the database.
- It is the most widely used architecture to design a DBMS.

Relational database

- This database is based on the relational data model, which stores data in the form of rows(tuple) and columns(attributes), and together forms a table(relation).
- A relational database uses SQL for storing, manipulating, as well as maintaining the data.
- Each table in the database carries a key that makes the data unique from others.
- Examples of Relational databases are MySQL, Microsoft SQL Server, Oracle, etc.



Properties of relational database

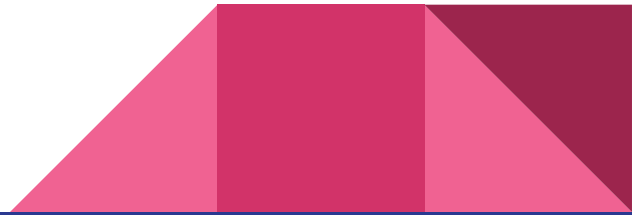
ACID :

Atomicity: This ensures the data operation will complete either with success or with failure. It follows the 'all or nothing' strategy.

Consistency: If we perform any operation over the data, its value before and after the operation should be preserved.

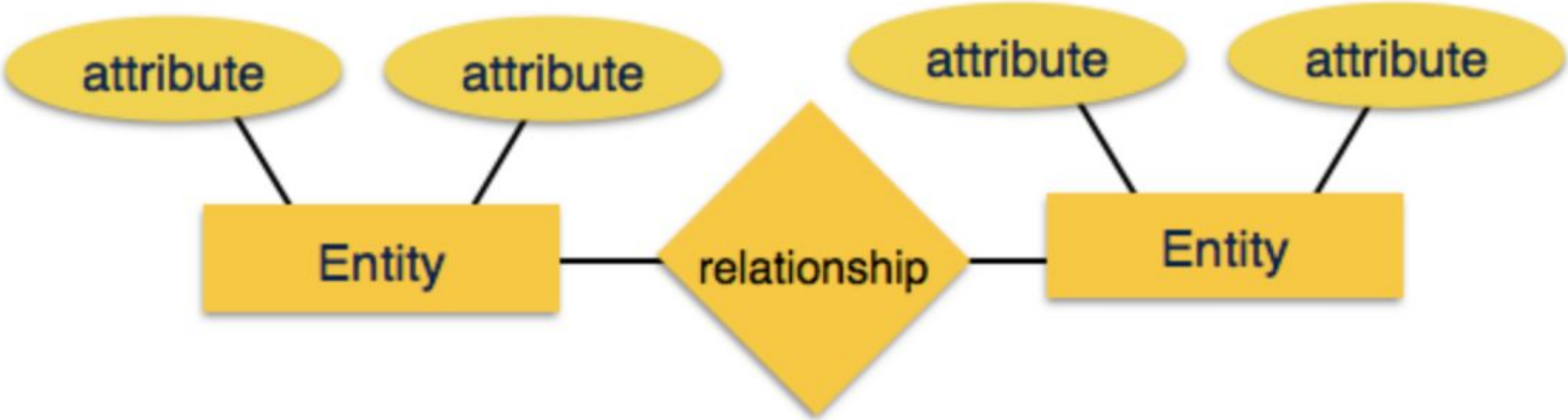
Isolation: There can be concurrent users for accessing data at the same time from the database. Thus, isolation between the data should remain isolated.

Durability: It ensures that once it completes the operation and commits the data, data changes should remain permanent.



Relational database

- Entity
- Attribute
- Relationship



Examples of Relational Database

School database

Students	Teachers	Subjects	Timetable	School
School id				
Roll no	Id	Id		
Age	Subject	Name		
Name	Class	Teachers {}		
Class	Employment status			
Address, parent name	Name, address, age			
Blood group	Email, contact			

Examples of Relational Database

Airline database

Customer info	Loyalty info	Aircraft details	Pilot info/ staff info	Schedule
Customer ID (Primary key, name	Frequent flyer no	Model no	Type (ground/crew/pilot)	Aircraft id
PNR {}	CustomerID	Serial no		From airport name
Contact no	class/category	Age of aircraft		To airport name
Frequent flyer no	Rewards obtained	Pilots {}		country
	Miles	Maintenance state		

Customer ID (primary key)

FFN (foreign key)

FFN (primary key)

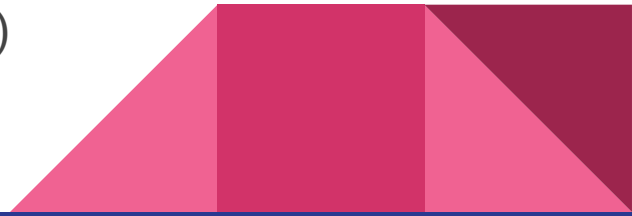
Scene 2 : CID <-> FFN

Customer ID (primary key)

FFN (foreign key)

FFN (primary key/foreign key)

Customer ID (foreign key)



Examples of Relational Database

Hospital database

Hospital	Patient	Doctor	Equipment
id	id	id	code
city	Hospital id	Hospital id	Safety check
address	DOB, all personal details	speciality	Hospital id
Departments {}	Doctor id	Personal details	quantity
		Emergency contact	
		License no	

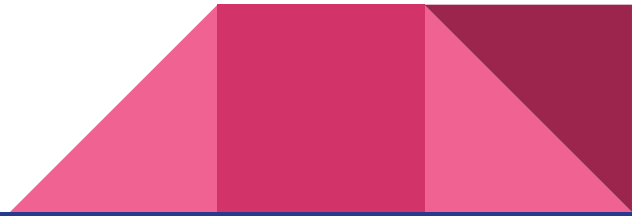
Examples of Relational Database

Banking database

Bank	Transaction	Customer	Loan	Staff
name	id	id	id	id
ifsc	amount	Personal details	type	name
City, state, branch	mode	Account type	Interest rate	department
	date	Loan type	T&C	Monthly salary
		Loan id	PoC	
		Has_loan (T/F)		

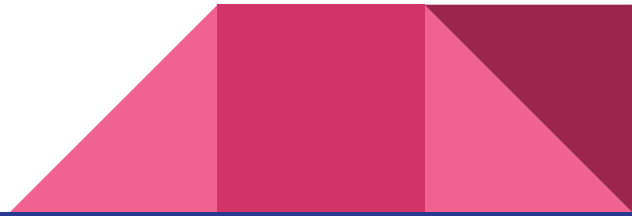
ER Model

- Entity-Relationship (ER) Model is based on the notion of real-world entities and relationships among them.
- While formulating real-world scenario into the database model, the ER Model creates entity set, relationship set, general attributes and constraints.
- ER Model is best used for the conceptual design of a database.



ER Model

- **Entity** − An entity in an ER Model is a real-world entity having properties called attributes. Every attribute is defined by its set of values called domain.
- **Relationship** − The logical association among entities is called relationship. Relationships are mapped with entities in various ways. Mapping cardinalities define the number of association between two entities.
- **Mapping cardinalities −**
  - one to one
  - one to many
  - many to one
  - many to many



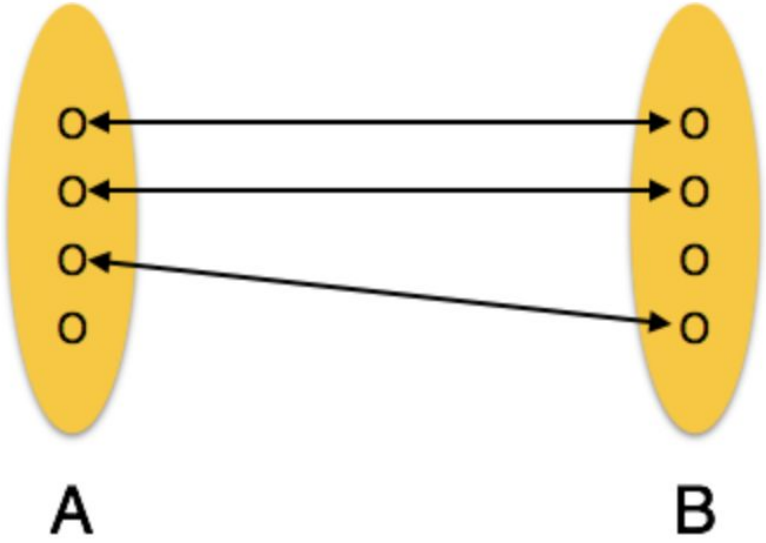
Attributes

Types of Attributes

- Simple attribute – Simple attributes are atomic values, which cannot be divided further. For example, a student's phone number is an atomic value of 10 digits.
- Composite attribute – Composite attributes are made of more than one simple attribute. For example, a student's complete name may have first_name and last_name.
- Derived attribute – Derived attributes are the attributes that do not exist in the physical database, but their values are derived from other attributes present in the database. For example, average_salary in a department should not be saved directly in the database, instead it can be derived. For another example, age can be derived from data_of_birth.
- Single-value attribute – Single-value attributes contain single value. For example – Social_Security_Number.
- Multi-value attribute – Multi-value attributes may contain more than one values. For example, a person can have more than one phone number, email_address, etc.

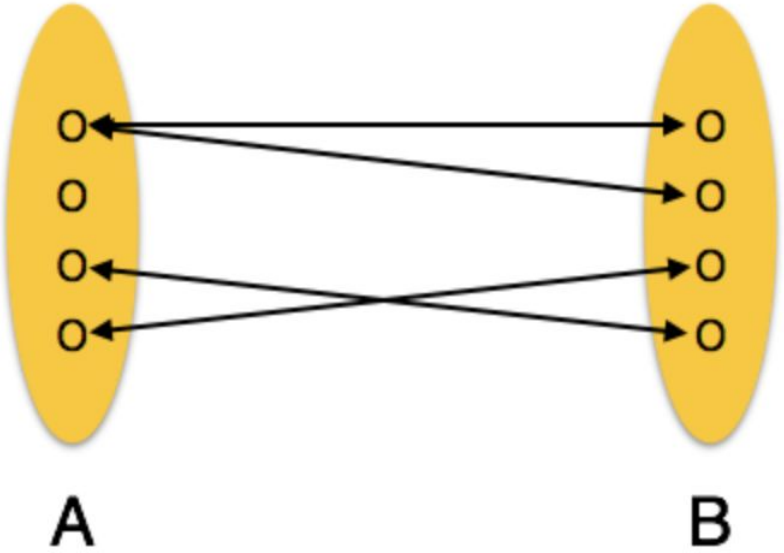
Mapping Cardinalities

- **One-to-one** − One entity from entity set A can be associated with at most one entity of entity set B and vice versa.



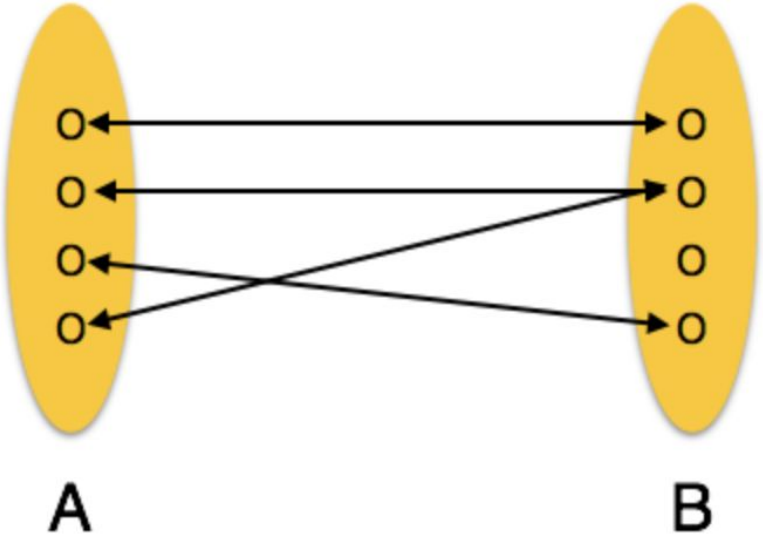
Mapping Cardinalities

- **One-to-many** − One entity from entity set A can be associated with more than one entities of entity set B however an entity from entity set B, can be associated with at most one entity.



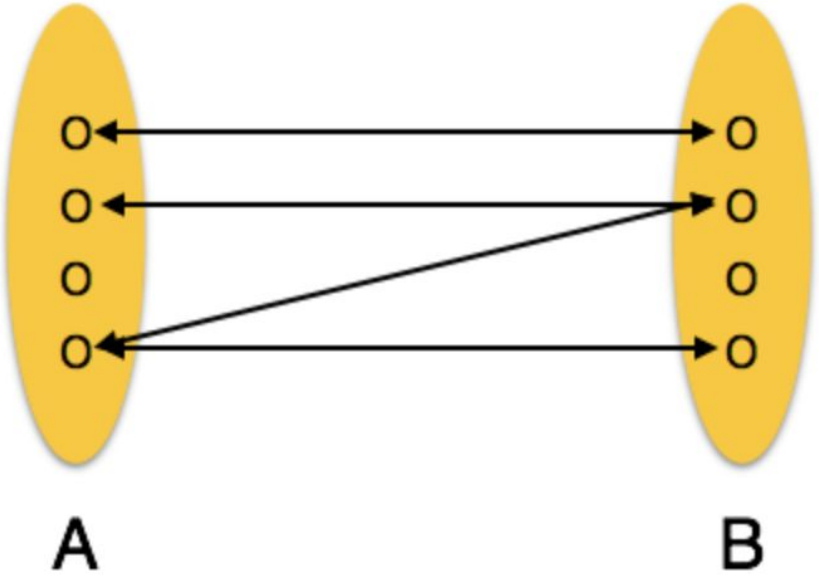
Mapping Cardinalities

- **Many-to-one** − More than one entities from entity set A can be associated with at most one entity of entity set B, however an entity from entity set B can be associated with more than one entity from entity set A.



Mapping Cardinalities

- **Many-to-many** − One entity from A can be associated with more than one entity from B and vice versa.



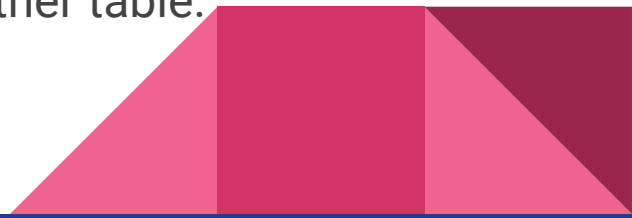
Keys

Primary Key:

A primary key is used to ensure data in the specific column is unique. It is a column cannot have NULL values. It is either an existing table column or a column that is specifically generated by the database according to a defined sequence.

Foreign Key:

A foreign key is a column or group of columns in a relational database table that provides a link between data in two tables. It is a column (or columns) that references a column (most often the primary key) of another table.



Keys

STUDENT

STUD_NO	STUD_NAME	STUD_PHONE	STUD_STATE	STUD_COUNTRY	STUD_AGE
1	RAM	9716271721	Haryana	India	20
2	RAM	9898291281	Punjab	India	19
3	SUJIT	7898291981	Rajsthan	India	18
4	SURESH		Punjab	India	21

Table 1

STUDENT_COURSE

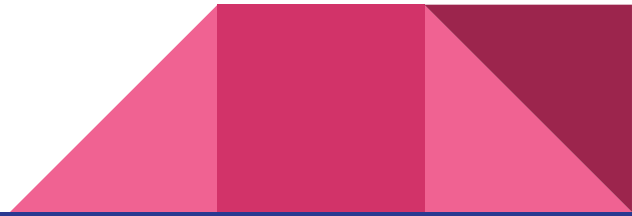
STUD_NO	COURSE_NO	COURSE_NAME
1	C1	DBMS
2	C2	Computer Networks
1	C2	Computer Networks

Table 2

S.NO.	PRIMARY KEY	FOREIGN KEY
1	A primary key is used to ensure data in the specific column is unique.	A foreign key is a column or group of columns in a relational database table that provides a link between data in two tables.
2	It uniquely identifies a record in the relational database table.	It refers to the field in a table which is the primary key of another table.
3	Only one primary key is allowed in a table.	Whereas more than one foreign key are allowed in a table.
4	It is a combination of UNIQUE and Not Null constraints.	It can contain duplicate values and a table in a relational database.
5	It does not allow NULL values.	It can also contain NULL values.
6	Its value cannot be deleted from the parent table.	Its value can be deleted from the child table.
7	It constraint can be implicitly defined on the temporary tables.	It constraint cannot be defined on the local or global temporary tables.

SQL overview

- SQL is a programming language for Relational Databases.
- SQL comprises both data definition and data manipulation languages.
- Using the data definition properties of SQL, one can design and modify database schema, whereas data manipulation properties allows SQL to store and retrieve data from database.



SQL overview

- The SQL programming language was first developed in the 1970s by IBM researchers Raymond Boyce and Donald Chamberlin.
- n 1979, a company called Relational Software, which later became Oracle, commercially released its own version of the SQL language called Oracle V2.
- Since then, the American National Standards Institute (ANSI) and the International Standards Organization (ISO) have deemed the SQL language the standard language in relational database communication.
- While major SQL vendors do modify the language to their desires, most base their SQL programs off of the ANSI approved version.

Questions?