Foundation University

Rawalpindi Campus

Introduction to Database Systems – CSC - 221

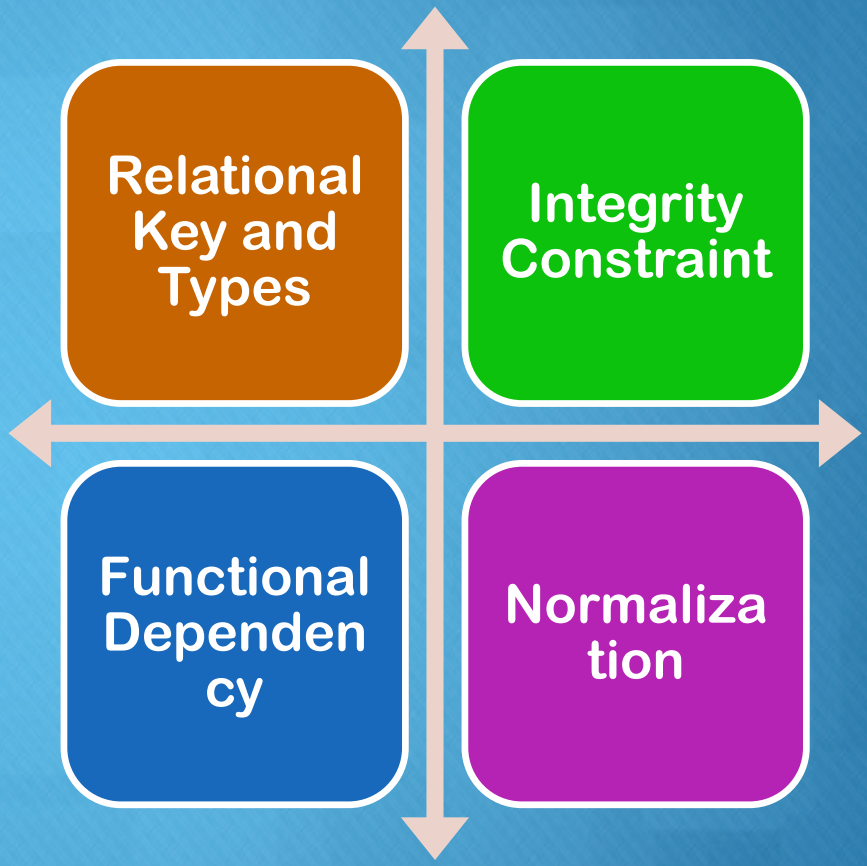
A Presentation by

R.M Hafeez Javed
Lecturer – Software Engineering
www.rmhjaved.com

RECAP

- MY NAME IS……
- I REMEMBER……

Objectives of Today's Lecture



Relational Key

- A **key** part of a **relational** database.
- They ensure each record within a table can be uniquely identified by one or a combination of fields within the table.
- They help enforce integrity and help identify the relationship between relations.

Need of the Key/Why Key

- They ensure that each record in a table is precisely identified.
- They help establish and enforce various types of integrity.
- Used to establish and identify relation between tables.
- Ensure that each record within a table can be uniquely identified by combination of one or more fields within a table.
- Defined in a table to access or sequence the stored data quickly and smoothly.

Types of Key

- Super key
- Candidate key
- Primary key
- Alternative key
- Foreign key
- Composite key

Super Key

- A super key is an attribute or combination of attributes in a relation that identifies a tuple uniquely within the relation.
- A super key is the most general type of key.
- For example, in a relation STUDENT consists of different attributes like RegistrationNo, Name, FatherName, Class and Address. The only attribute that can uniquely identify a Tuple in a relation is RegistrationNo.
- The Name attribute cannot identify a tuple because two or more students may have the same Name.
- It means that RegistrationNo is the super key for the relation.
- Any combination of attributes with the super key is also a super key.

Candidate Key

- A candidate key is a super key that contains no extra attribute.
- It consists of minimum possible attributes. A super key like {RegistrationNo, Name} contains an extra field Name. It can be used to identify a tuple uniquely in the relation, But it does not consist of minimum possible attribute as only RegistrationNo can be used to identify a tuple in a relation.
- It means that {RegistrationNo, Name} is a super key but it is not a candidate key because it contains an extra field. On the other hand, RegistrationNo is a super key as well as candidate key.

Primary Key

- A primary key is a candidate key that is selected by the database designer to identify tuples uniquely in a relation. A relation may contain many candidate keys. When the designer selects one of them to identify a tuple in the relation, it becomes a primary key.
- A relation can have only one primary key, each value in primary key attribute must be unique and can not contain null values.
- Suppose a relation STUDENT contains different attributes such as RegNo, Name and Class. The attribute RegNo uniquely identifies each student in a table. It can be used as primary key for this table. The attribute Name can not uniquely identify each row because two students can have same names. It can not be used as a primary key.

Alternative Key

- The candidate keys that are not selected as primary key are known as alternate keys.
- Suppose STUDENT relation contains different attributes such as RegNo, RollNo, Name and Class. The attributes RegNo and RollNo can be used to identify each student in the table.
- If Regno is selected as primary key then RollNo attribute is known as alternate key.

Composite Key

- A primary key that consists of two or more attributes is known as composite key.

Foreign Key

- A foreign key is an attribute or set of attributes in a relation whose values match a primary key in another relation.
- The relation in which foreign key is created is known as Dependent Table or Child Table. The relation to which the foreign key refers is known as Parent Table.

Consider table EMP and DEPT

EMP (empId, empName, qual, depId)

DEPT (depId, depName, numEmp)

Non-key Attributes

- **Non-key** attributes are attributes other than **candidate key** attributes in a table.
- EMP (empId, empName, qual, depId)

Integrity Constraints

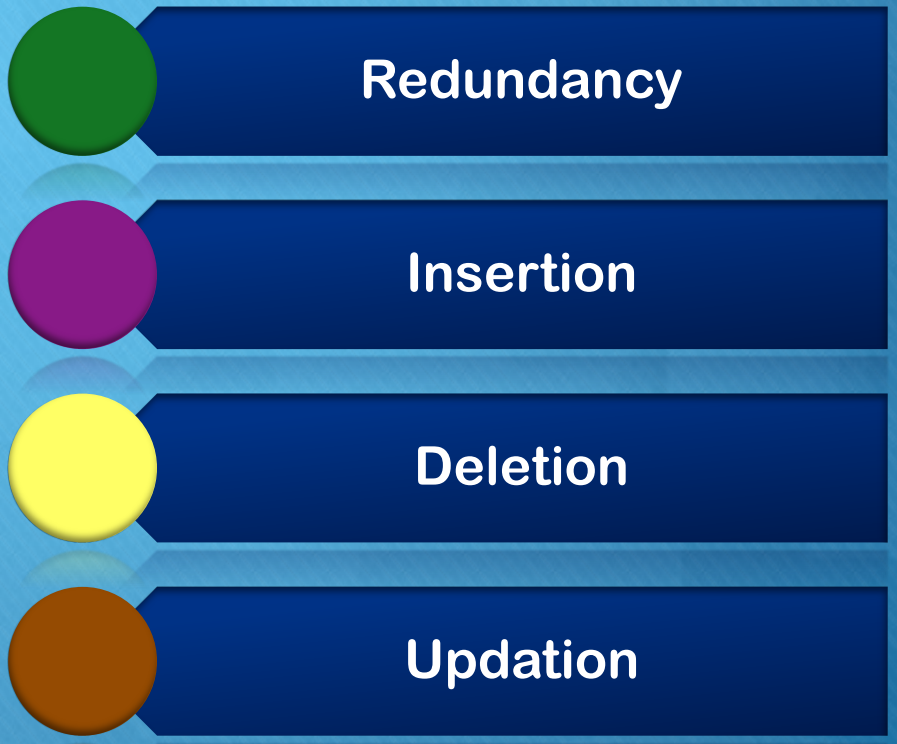
- Entity integrity
  - Primary key cannot have null value.
- Referential integrity
  - If a foreign key exists in a relation, either the foreign key value must match a candidate key value of some tuple in its home relation or the foreign key value must be wholly null.

Normalization

- A step by step process to produce more efficient and accurate database design.
- Purpose is to produce an anomaly free design that is smaller and well-structured.

Anomalies

- An inconsistent, incomplete or incorrect state of database.
- Four types of anomalies are of concern here.



Functional Dependency

- Normalization is based on the concept of functional dependency. A functional dependency is a type of relationship between attributes.
- If A and B are attributes or sets of attributes of relation R, we say that B is functionally dependent on A if each value of A in R, associated with exactly one value of B in R.
- We write this as A $\longrightarrow$ B, read as “A functionally determines B” or “ A determines B”.

Functional Dependency…

- STD ⟶ (stId, stName, stAdr, prName, credits)
  stId ⟶ stName, stAdr, prName, credits
  prName ⟶ credits

- Attribute on the left side of the arrow is called a **determinant** and on right are called **dependents**.

Goals of Normalization

- Eliminate redundant data (for example, storing the same data in more than one table).
- Ensure data dependencies make sense (only storing related data in a table).
- Both of these are worthy goals as they reduce the amount of space a database consumes, and ensure that data is logically stored.

Normal Forms

- First Normal Form
- Second Normal Form
- Third Normal Form
- Boyce - Codd Normal Form
- Higher Forms

First Normal Form

- A relation is in first normal (1st NF) form if and only if
- Every attribute is single valued for each tuple.
- This means that each attribute in each row , or each cell of the table, contains only one value.
- No repeating fields or groups are allowed.

Second Normal Form

- A relation is in second normal form (2NF) if and only if
- It is in first normal form (1st NF) and All the nonkey attributes are fully functionally dependent on the key.
- The only time, we have to be concerned about 2NF, when the key is composite.
- Removing the partial dependency

Third Normal Form

- A relation is in Third Normal Form (3rd NF) if and only if
- It is in First (1st NF) and Second Normal Form (2nd NF) and in which no non-primary-key attribute is dependent on another non-key attribute.
- All non-key attributes are functionally dependent only on primary key.
- **Remove transitive dependency**

CHHUTTI

AND THAT IS FAREWELL TO DAY 16-17 ☺