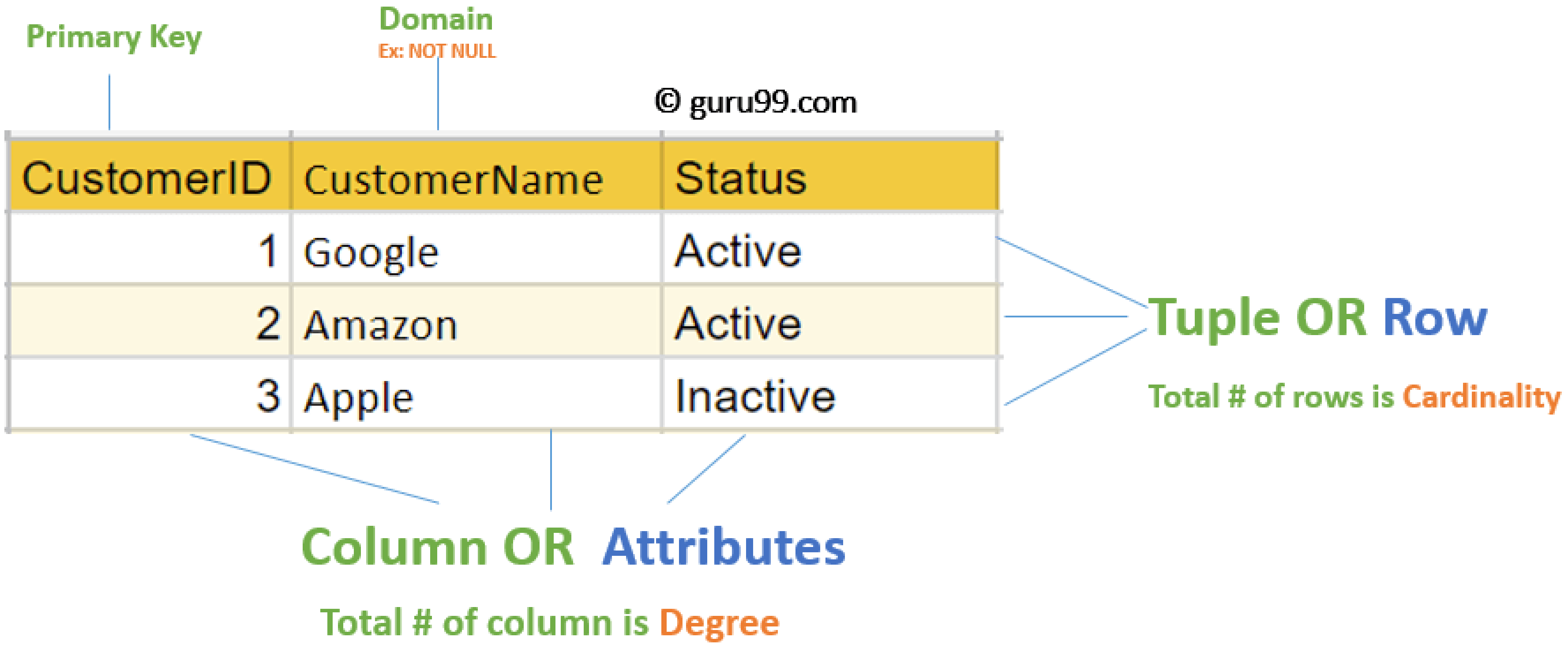
Relational Database Management System

- Collection of information organized in tables
  - *Tables are also "relations"*
- Tables are constructed and associated to each other through shared fields–"common" fields
  - *Fields are also "columns" or "attributes"*
- A set of attributes comprises a record
  - *Records are also "rows" or "tuples"*
- Tables are related through common fields designated as primary and foreign keys
- Allow us to find, update, and delete data quickly, and help to ensure accuracy

Table also called Relation



Primary Key

Domain

Ex: NOT NULL

© guru99.com

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive

Tuple OR Row

Total # of rows is Cardinality

Column OR Attributes

Total # of column is Degree

Terminologies

Domain: It contains a set of atomic values that an attribute can take.

Attribute: It contains the name of a column in a particular table. Each attribute Ai must have a domain, dom(Ai)

Relational instance: In the relational database system, the relational instance is represented by a finite set of tuples. Relation instances do not have duplicate tuples.

Relational schema: A relational schema contains the name of the relation and name of all columns or attributes.

Relational key: In the relational key, each row has one or more attributes. It can identify the row in the relation uniquely.

Cardinality: Total number of tuples present in a relation.

Degree of Relation: No. of attributes in a relational schema

Properties of Relations

- Name of the relation is distinct from all other relations.
- Each relation cell contains exactly one atomic (single) value
- Each attribute contains a distinct name
- Attribute domain has no significance
- tuple has no duplicate value
- Order of tuple can have a different sequence

Differences Between a Database and a Relational Database

- Relational database stores data in a tabular form -- or arranged in a table with rows and columns -- while a database stores data as files.
- Database normalization is present in a relational database while it is not present in a database.
- A relational database supports a distributed database while a database does not support a distributed database.
- In a relational database, the data values are stored in the form of tables and each table possesses a primary key. In a database, data is normally stored in hierarchical or navigational form.
- Since data is stored in the form of tables in a relational database, then the relationship between these data values is stored as well. Since a database stores data as files, then there is not relationship between the values or tables.
- In a relational database, the integrity constraints are defined for the purpose of an ACID On the other hand, a database does not utilize any security to protect against data manipulation.
- While a relational database is designed to support large amounts of data and multiple users, a database is designed to deal with small amounts of data and one single user.

Advantages of Relational Databases

Simple Model:

- The simplest model of the relational database does not require any complex structure or query to process the databases.
- It has a simple architectural process as compared to a hierarchical database structure.
- Its simple architecture can be handled with simple SQL queries to access and design the relational database.

Data Accuracy:

- Relational databases can have multiples tables related to each other through primary and foreign keys.
- There are fewer chances for duplication of data fields.
- Therefore the accuracy of data in relational database tables is greater than in any other database system.

Easy to access Data:

- The data can be easily accessed from the relational database, and it does not follow any pattern or way to access the data.
- One can access any data from a database table using SQL queries.
- Each table in the associated database is joined through any relational queries such as join and conditional descriptions to concatenate all tables to get the required data.

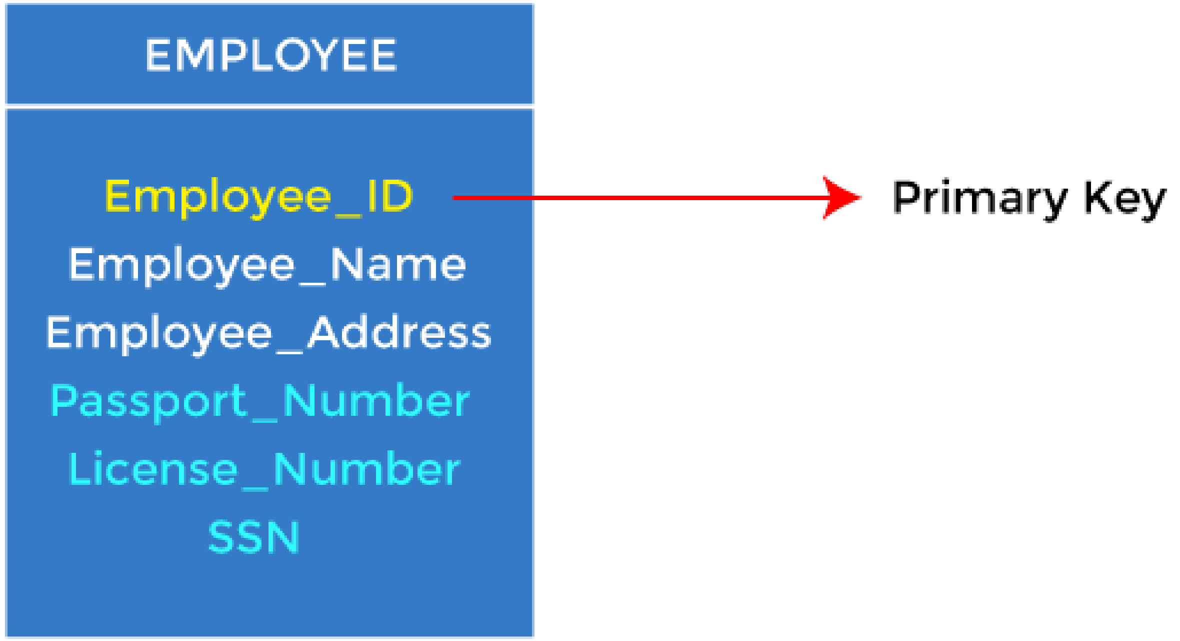
Security: It sets a limit that allows specific users to use relational data in RDBMS.

Collaborate: It allows multiple users to access the same database at a time.

Types of Keys

Primary Key

- It is the first key used to identify one and only one instance of an entity uniquely.
- An entity can contain multiple keys.
- The key which is most suitable from those lists becomes a primary key.
- In the EMPLOYEE table, ID can be the primary key since it is unique for each employee.
- In the EMPLOYEE table, we can even select License_Number and Passport_Number as primary keys since they are also unique.
- For each entity, the primary key selection is based on requirements and developers.



EMPLOYEE

Employee_ID

Employee_Name

Employee_Address

Passport_Number

License_Number

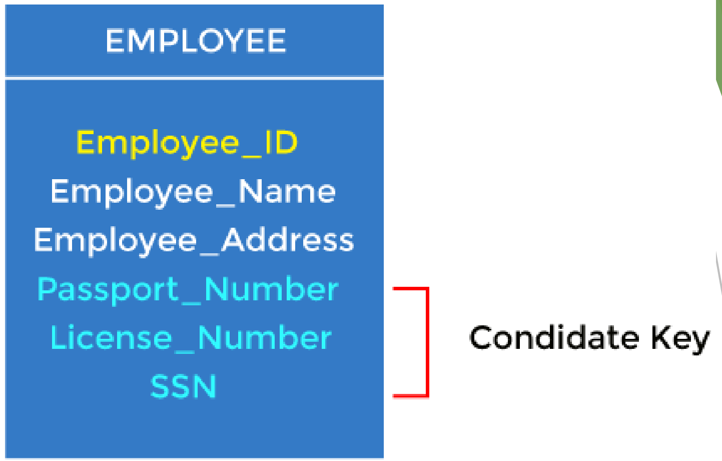
SSN

Primary Key

Types of Keys

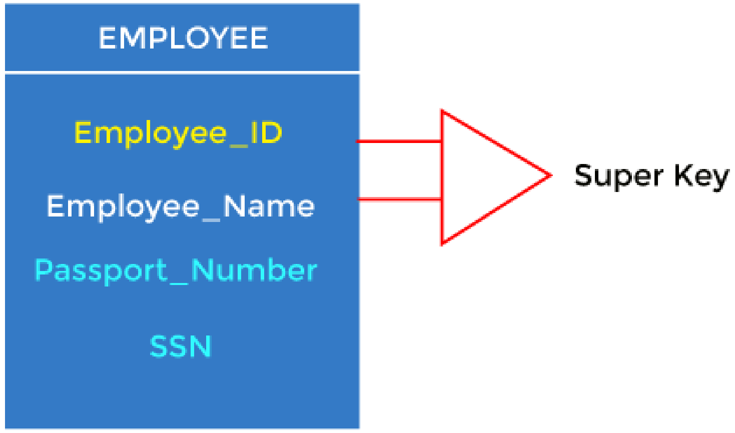
Candidate Key

- A candidate key is an attribute or set of attributes that can uniquely identify a tuple.
- Except for the primary key, the remaining attributes are considered a candidate key. The candidate keys are as strong as the primary key.



Super Key

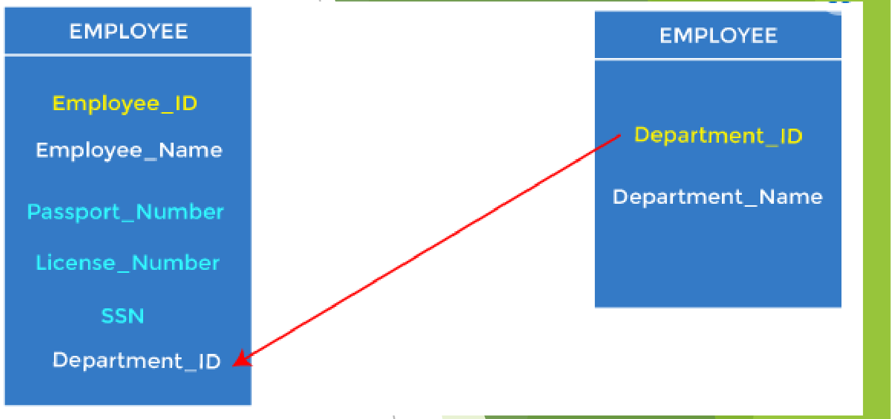
- Super key is an attribute set that can uniquely identify a tuple.
- A super key is a superset of a candidate key.



Types of Keys

Foreign Key

- Foreign keys are the column of the table used to point to the primary key of another table.



EMPLOYEE

Employee_ID

Employee_Name

Passport_Number

License_Number

SSN

Department_ID

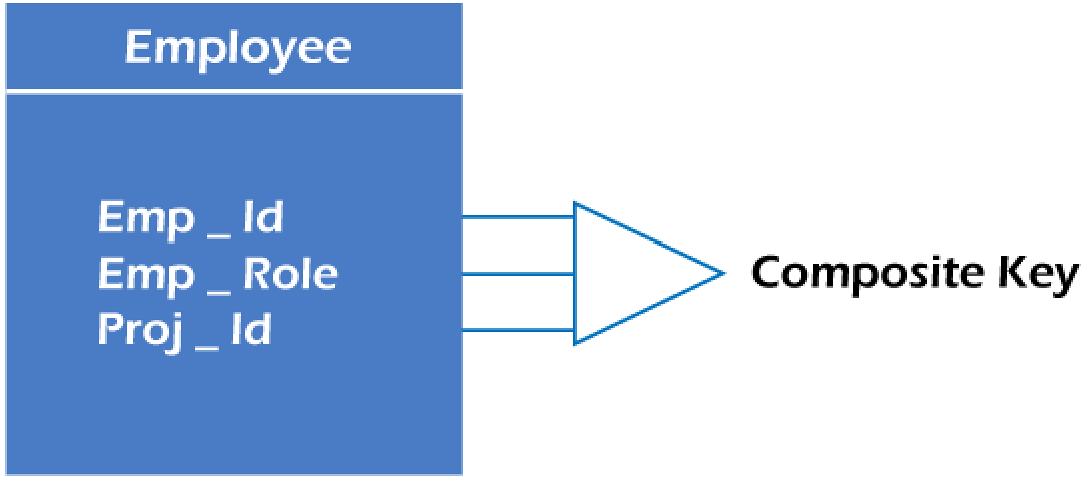
EMPLOYEE

Department_ID

Department_Name

Composite Key

- Whenever a primary key consists of more than one attribute, it is known as a composite key.
- This key is also known as Concatenated Key.



Employee

Emp _ Id

Emp _ Role

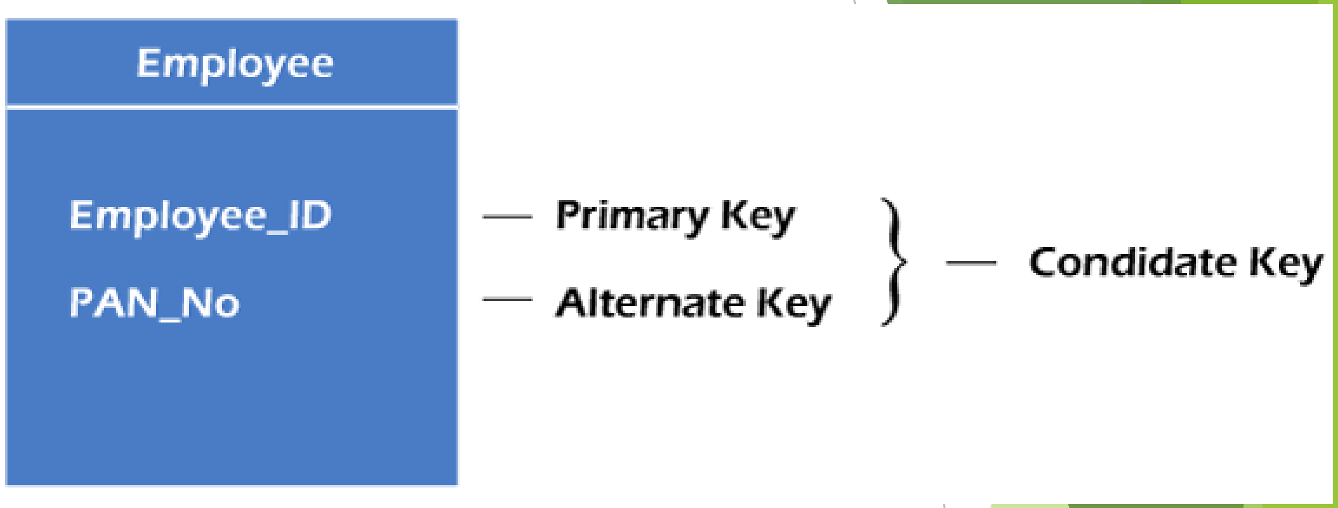
Proj _ Id

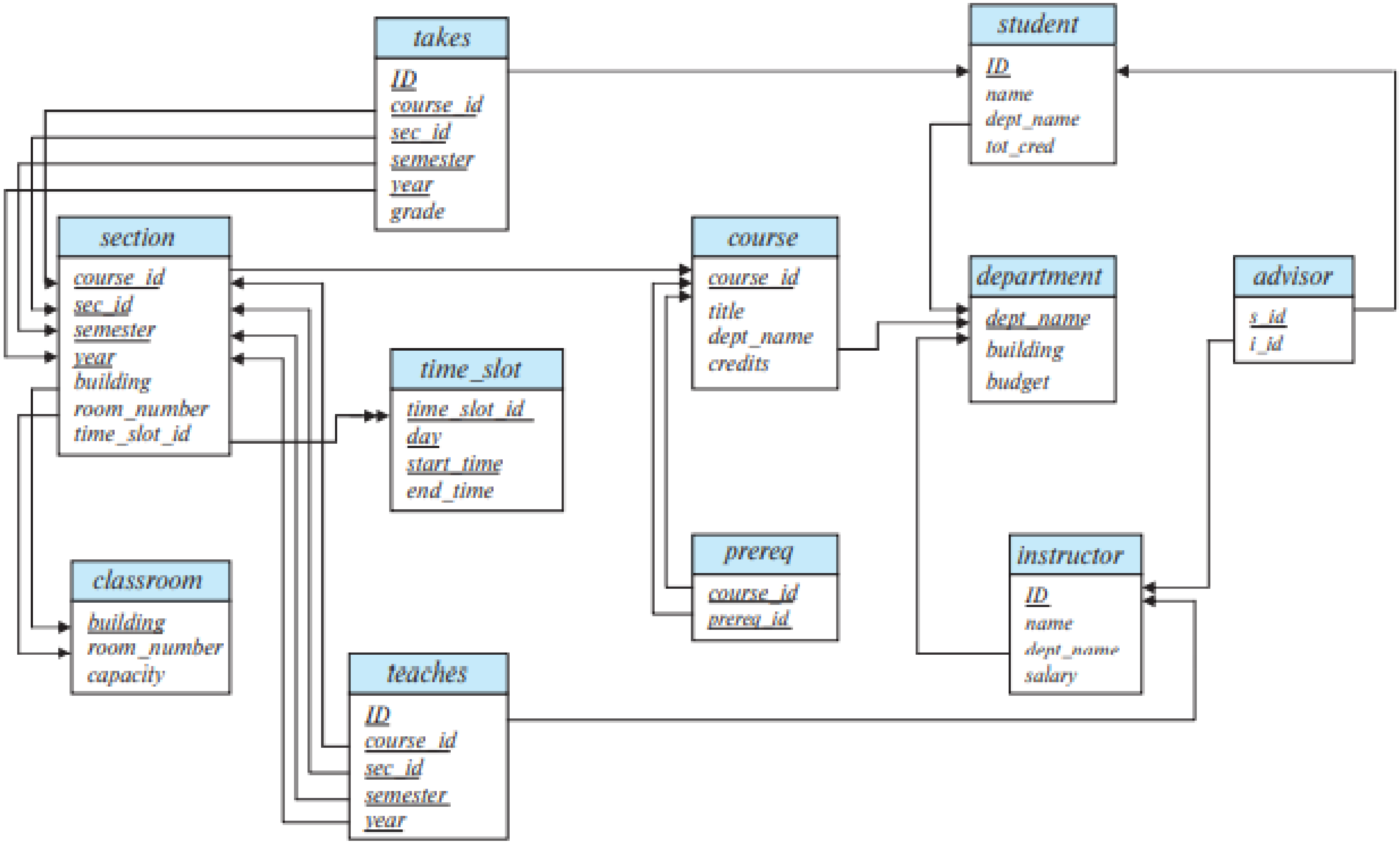
Composite Key

Types of Keys

Alternate Key

- There may be one or more attributes or a combination of attributes that uniquely identify each tuple in a relation.
- These attributes or combinations of the attributes are called the **candidate keys**.
- One key is chosen as the primary key from these candidate keys, and the remaining candidate key, if it exists, is termed the alternate key.
- **In other words,** the total number of the alternate keys is the total number of candidate keys minus the primary key.
- The alternate key may or may not exist. If there is only one candidate key in a relation, it does not have an alternate key.





takes

ID
course_id
sec_id
semester
year
grade

student

ID
name
dept_name
tot_cred

section

course_id
sec_id
semester
year
building
room_number
time_slot_id

course

course_id
title
dept_name
credits

department

dept_name
building
budget

advisor

s_id
i_id

time_slot

time_slot_id
day
start_time
end_time

classroom

building
room_number
capacity

prereq

course_id
prereq_id

instructor

ID
name
dept_name
salary

teaches

ID
course_id
sec_id
semester
year

What is Normalization?

- The process by which we efficiently organize data to achieve these goals:
  - Eliminating redundancy
  - Ensuring data is stored in the correct table
  - Eliminating need for restructuring database when data is added.
- Five levels of normal form
  - In order to achieve one level of normal form, each previous level must be met

Third normal form is sufficient for most typical database applications.

First Normal Form (1NF)

- There are no repeating or duplicate fields.
- Each cell contains only a single value.
- Each record is unique.
  - *Identified by primary key*

Example

item	colors	price	tax
T-shirt	red, blue	12.00	0.60
polo	red, yellow	12.00	0.60
T-shirt	red, blue	12.00	0.60
sweatshirt	blue, black	25.00	1.25

Table is not in first normal form because:

- Multiple items in color field
- Duplicate records / no primary key

Example

item	color	price	tax
T-shirt	red	12.00	0.60
T-shirt	blue	12.00	0.60
polo	red	12.00	0.60
polo	yellow	12.00	0.60
sweatshirt	blue	25.00	1.25
sweatshirt	black	25.00	1.25

Table is now in first normal form.

Second Normal Form (2NF)

- All non-key fields depend on all components of the primary key.
  - *Guaranteed when primary key is a single field.*

Example

item	color	price	tax
T-shirt	red	12.00	0.60
T-shirt	blue	12.00	0.60
polo	red	12.00	0.60
polo	yellow	12.00	0.60
sweatshirt	blue	25.00	1.25
sweatshirt	black	25.00	1.25

Table is not in second normal form because:

- **price** and **tax** depend on **item**, but not **color**

Example

item	color
T-shirt	red
T-shirt	blue
polo	red
polo	yellow
sweatshirt	blue
sweatshirt	black

item	price	tax
T-shirt	12.00	0.60
polo	12.00	0.60
sweatshirt	25.00	1.25

Tables are now in second normal form.

Third Normal Form (3NF)

- No non-key field depends upon another.
  - All non-key fields depend only on the primary key.

Example

item	color
T-shirt	red
T-shirt	blue
polo	red
polo	yellow
sweatshirt	blue
sweatshirt	black

item	price	tax
T-shirt	12.00	0.60
polo	12.00	0.60
sweatshirt	25.00	1.25

Tables are not in third normal form because:

- **tax** depends on **price**, not **item**

Example

item	color
T-shirt	red
T-shirt	blue
polo	red
polo	yellow
sweatshirt	blue
sweatshirt	black

item	price
T-shirt	12.00
polo	12.00
sweatshirt	25.00

price	tax
12.00	0.60
25.00	1.25

Tables are now in third normal form.

Another Example

Name	Assignment 1	Assignment 2
Jeff Smith	Article Summary	Poetry Analysis
Nancy Jones	Article Summary	Reaction Paper
Jane Scott	Article Summary	Poetry Analysis

Another Example

Assignment ID	Description
1	Article Summary
2	Poetry Analysis
3	Reaction Paper

Student ID	First Name	Last Name
1	Jeff	Smith
2	Nancy	Jones
3	Jane	Scott

Assignment ID	Student ID
1	1
1	2
1	3
2	1
2	3
3	2

Tables are in Third Normal Form

Relationships

- Relationships are created between tables using the primary key field and a foreign key field
  - One to One Relationship
    - One record in a table relates to one record in another table
  - One to Many Relationship
    - One record in a table can relate to many records in another table
  - Many to Many Relationship
    - Many records in one table can relate to many records in another table

Relational Integrity constraints

- Relational Integrity constraints is referred to conditions which must be present for a valid relation.
- These integrity constraints are derived from the rules in the mini-world that the database represents.
- There are many types of integrity constraints.
  - ✓ Inherent Model based
  - ✓ Schema Based
  - ✓ Application Based
- Schema Based Constraints on the Relational database management system is mostly divided into three main categories are:

1. Domain constraints
2. Key constraints
3. Referential integrity constraints

Domain Constraints

- Domain constraints can be violated if an attribute value is not appearing in the corresponding domain or it is not of the appropriate data type.
- Domain constraints specify that within each tuple, and the value of each attribute must be unique.
- This is specified as data types which include standard data types integers, real numbers, characters, Booleans, variable length strings, etc.

Key constraints

- An attribute that can uniquely identify a tuple in a relation is called the key of the table.
- The value of the attribute for different tuples in the relation has to be unique.

Referential integrity constraints

- Referential integrity constraints is base on the concept of Foreign Keys.
- A foreign key is an important attribute of a relation which should be referred to in other relationships.
- Referential integrity constraint state happens where relation refers to a key attribute of a different or same relation.
- However, that key element must exist in the table.

CODD’s 12 Rule

Codd’s 12 rules are a set of thirteen rules (numbered zero to twelve) proposed by Edgar F. Codd.

Rule 0: The system has to qualify as Relational, as a Database, and also as a Management System.

Rule 1: The information rule: Each and every information in the database is to be represented uniquely, mainly name values in column positions within a different row of a table.

Rule 2: The guaranteed access rule: All data must be ingressive. It says that every scalar value in the database must be correctly/logically addressable.

Rule 3: Systematic treatment of null values: The DBMS must allow each tuple to remain null.

Rule 4: Active online catalog (database’s structure) based on the relational model: The system must support an online, relational etc. structure which is ingressive to allowed users by means of their regular query.

Rule 5: The comprehensive data sublanguage: The system has to assist a minimum of one relational language that:

1. Has a linear syntax
2. Which can be used as both interactively and within application programs,
3. It supports data definition operations(DDL), data manipulation operations(DML), security and integrity constraints, and transaction management operations (begin, commit, and rollback).

CODD’s 12 Rule

Rule 6: The view updating rule: All views that theoretically improve must be upgradable by the system.

Rule 7: High-level insert, update, and delete: The system must support insert, update, and delete operators.

Rule 8: Physical data independence: Modify the physical level (how the data is stored, using arrays or linked lists etc.) must not require a modification to an application.

Rule 9: Logical data independence: Modify the logical level (tables, columns, rows etc.) must not require a modification to an application.

Rule 10: Integrity independence: Integrity constraints must be identified individually from application programs and stored in the catalog.

Rule 11: Distribution independence: The distribution of portions of a database to different locations should not be visible to users of the database.

Rule 12: The nonsubversion rule: If the system provides a low-level (i.e. records) interface, then that interface can’t be used to subvert the system.

Disadvantages of using Relational model

•Few relational databases have limits on field lengths which can't be exceeded.

•Relational databases can sometimes become complex as the amount of data grows, and the relations between pieces of data become more complicated.

•Complex relational database systems may lead to isolated databases where the information cannot be shared from one system to another.