Table of Content

[1. Relation Database Model:- 2](#_Toc30022)

[1.1. Relational Model Concepts in DBMS 3](#_Toc27749)

[1.2. Relational Database Design 4](#_Toc28387)

[1.3. Disadvantages Of Relational Databases 4](#_Toc29514)

[2. Logistic view of data keys and integrity model 5](#_Toc14369)

[2.1. Domain Integrity 5](#_Toc21099)

[2.2. Entity integrity 6](#_Toc27727)

[2.3. Referential integrity 6](#_Toc15050)

[2.4. Referential integrity in Microsoft Access 7](#_Toc14227)

[2.4.1. Foreign key rules 8](#_Toc4731)

[2.4.2. Enterprise Constraints 8](#_Toc12331)

[3. Normalization of Database 8](#_Toc9556)

[3.1. Normalization Rule 9](#_Toc3805)

[3.2. Problems Without Normalization 9](#_Toc7204)

[3.3. Normalization Rule 10](#_Toc26583)

[3.3.1. First Normal Form (1NF) 11](#_Toc30358)

[3.3.2. Second Normal Form (2NF) 11](#_Toc20357)

[3.3.3. Third Normal Form (3NF) 11](#_Toc25892)

[3.3.4. Fourth Normal Form (4NF) 12](#_Toc29485)

[4. Why do we need a Relational Database Model? 12](#_Toc3028)

[4.1. What is the Concept of Relational Database? 13](#_Toc11084)

[4.2. Relational Database Work 13](#_Toc15241)

[4.3. Relational Database Model's Structure 13](#_Toc32273)

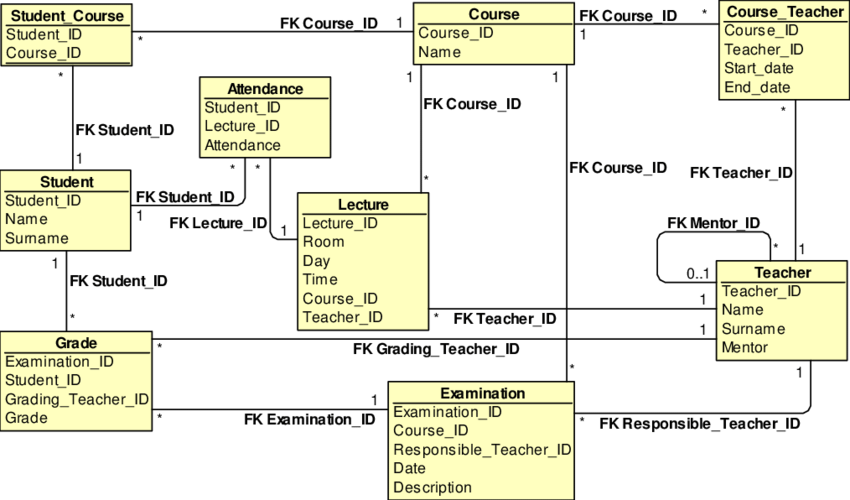
[5. References: 15](#_Toc24367)

# 1. Relation Database Model:-

**Relational Model (RM)** data is shown as a set of connections in a database. Simply said, a relation is a data table. Each set of values in a row in the table is an example of a data set. Each row in this table represents a concrete thing or connection in the actual world.

Each row's values may be better understood with the aid of the table and column names. The information is represented as a network of connections. Records are organised into tables in the relational paradigm. However, the data's logical organisation has no bearing on how the data are stored physically.

* Tables are used to display all data (relations)
* There are rows and columns in a table (tuples)
* (Officially) the rows are not arranged (i.e., the order in which rows are referenced does not matter)
* Duplicate rows are not allowed in a good relational table.
* A primary key is the principal identifier in a database table, and it may be made up of any number of fields.
* A table is connected to another by adding the primary key of the connected table, which is often a single column (like OWNERNUM for OWNERS). A foreign key is a special kind of column used in databases.
* Authentically reflects the problem's underlying structure
* Provides a complete representation of future data
* prevents information from being stored twice
* Facilitates quick data retrieval
* Helps guarantee that information stays as accurate as possible throughout time
* Very well organised and straightforward



**Some popular Relational Database management systems are:**

* IBM's Dynamic Server DB2 and Informix
* SQL Server and Access (Microsoft) and Oracle (Oracle) for Relational Databases

## Relational Model Concepts in DBMS

1. **In a table, each individual column is an attribute. Relationships may be described by their attributes. Students' Roll Numbers, Names, etc.**
2. Relations in the Relational model are stored in tabular form. Together with its entities, it is kept in storage. Both rows and columns are necessary components of a table. Records are shown in rows, whereas characteristics are displayed in columns.
3. Tuple is nothing more than a single row of data in a table.
4. The name of the connection and all of its characteristics are represented by the relation schema.
5. In a relation, the total number of qualities is referred to as the degree.
6. Cardinality refers to the total number of rows in a Table.
7. The values for a certain attribute are shown in a column.
8. An RDBMS relation instance is a discrete collection of tuples. Tuples in a relational instance are unique.
9. Each entry in a database table contains an identifier, or relation key, which may be a single value or a collection of values.
10. Attribute domain refers to the set of values and application areas for an attribute.

## Relational Database Design

* E-R Modeling, or Entity-Relationship Modeling, is a technique used in database design.
* This article has a condensed version.
* Using entities with characteristics to represent the data.
* What we call "things" or "ideas" might be grouped together into a category called "entities."
* Interdependencies between entities exist.
* The finished product is a normalised database that is easy to navigate and free of redundant entries.
* This ER diagram represents subsets of the 'parcels' sample database. Oracle data tables are imported into Microsoft Access, where they are then visualised using MS-Tool/Relationships Access's feature.
  1. **Disadvantages Of Relational Databases**

**Structure.** Since columns must be specified and data must accurately fit into rather strict categories, relational databases need a considerable degree of structure and forethought. Although the structure has its usefulness, it also presents challenges owing to factors including its high maintenance requirements and limited adaptability and scalability.

**Faulty upkeep, as expected.** As more data is added to the database, developers and other database staff need to spend time monitoring and expanding the database.

**Inflexibility.** Large volumes of unstructured data are difficult for relational databases to manage. Relational databases are not well-suited for data that is mostly qualitative, difficult to describe, or dynamic since the schema must grow along with the data, which is a time-consuming process.

**Unsuitable for large-scale use.** However, horizontally scaling relational databases across clusters of computers is a challenging problem. Managing relational databases across numerous servers is challenging due to the fact that the structure of a data set is disturbed as it becomes bigger and more spread, and the usage of several servers has an effect on performance (such as application response times) and availability.

# Logistic view of data keys and integrity model

The constraints of a relational model are essential. A well-defined notion of constraints on attributes or tables is supported by the relational model. In a database, constraints are helpful because they enable the designer to define how the data should be interpreted. The restrictions imposed by constraints ensure that DBMSs verify that the data meets the semantic requirements. Several distinct types of integrity restrictions are outlined below.

## Domain Integrity

When using a relational model, domain is a restriction that limits the possible values for attributes. On the other hand, domain constraints alone are insufficient to specify the whole range of data's possible meanings in the actual world. There should be clearer methods to specify which data values are acceptable and in what format an attribute should be stored. For instance, the Employee ID (EID) or birth date of each employee must be within the period [January 1, 1950, January 1, 2000]. Integrity constraints give this data in the form of logical assertions.

## Entity integrity

Every table must have a primary key in order to guarantee entity integrity. All PK values must be non-null, and the PK itself must not have any null values. This is due to the fact that we cannot recognise certain rows due to the presence of null values for the main key. Some users in the EMPLOYEE table, for instance, may not have access to a phone, thus that field can't serve as the primary key.

## Referential integrity

A foreign key cannot be null and must either correspond to a main key or be empty. Specified between two tables (parent and child), this constraint ensures that rows in both tables continue to correlate with one another. This indicates that the row reference must exist in both the source and target tables.

Some constraints on referential integrity in the Company's Customer/Order database:

* Customer(**CustID**, CustName)
* Order(**OrderID**, CustID, OrderDate)

Referential integrity must be enforced to prevent the creation of orphan records. If the value of a foreign key (FK) does not appear in the entity that houses the primary key (PK), then that record is considered an orphan. You may recall that most joins occur between a PK and FK.

The customer ID (CustID) in the Order table must correspond to a real CustID in the Customer record, as required by the referential integrity requirement. Deterministic referential integrity is standard in the relational database type. Referential integrity restrictions are defined at the time when the tables are created.

Another example from a database of courses and classes:

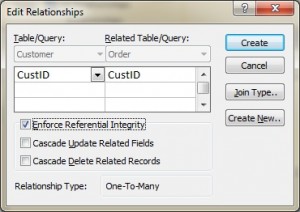
* Course(**CrsCode**, DeptCode, Description)
* Class(**CrsCode, Section**, ClassTime)

It is critical to have a valid CrsCode in both the Class and Course columns to ensure the database's referential integrity. The Class table's CrsCode and Section columns together produce the PK, therefore referential integrity must be strictly maintained.

The primary key (PK) and foreign key (FK) must have the same data type and belong to the same domain for the join to succeed when establishing referential integrity in a relational database management system. Popular relational database management systems are based on the relational model developed by E. F. Codd at IBM's San Jose Research Laboratory (RDBMS). In comparison to their forebears, modern relational database management systems are more simpler and easier to use.

## Referential integrity in Microsoft Access

The primary key (PK) of the Customer table is linked to the CustID of the Order table in order to provide referential integrity in MS Access. Refer to Figure 9.1 to see this in action on MS Access' Edit Relationships screen.



### Foreign key rules

Establishing referential integrity may need the addition of foreign key rules, such as the behaviour of dependent rows (in the Orders database) when the record containing the PK, part of the parent (Customer) is deleted or amended (updated). In MS Access, the Edit Relationships box has two more choices for FK rules (see Figure 9.1). Cascade Delete and Cascade Update may be accessed using these links. The system will not permit the deletion or update of the primary key (PK) of a parent record in the Customers database if that record currently exists. A PK and a record might be considered a parent-child pair.

With some databases, you'll also see Set to Null beside Delete. To do this, the PK row will be deleted and the foreign key in the child table will be set to NULL. There will be an orphan row as a consequence, but that's OK.

### Enterprise Constraints

* Enterprise constraints, also known as semantic constraints, allow users or database administrators to impose extra rules that are based on a large number of tables.
* Here are a few cases in point.
* There should be no more than 30 kids in one classroom.
* Each instructor is limited to four courses each semester.
* An employee is limited to a maximum of five projects at once.
* An employee's pay can't be higher than that of their supervisor.

# Normalization of Database

* What we call "database normalisation" is really a method for making sense of all the information stored inside. Data redundancy (repetition) and undesired qualities like Insertion, Update, and Deletion Anomalies may be removed using a process called normalisation, which involves the methodical dissection of tables. The method involves many steps and results in clean, tabular data that has had any duplicates removed from the relational databases. There are primarily two applications for normalisation, both of which focus on removing superfluous information.
* Making sure data storage makes sense, or that data dependencies make sense.

## Normalization Rule

* There are many distinct normal forms that govern normalisation:
* Normal Form, Initial
* Forma normale secunda
* Normality, Third Form
* Form Four BCNF
* Some primers:
* Constraints in Structured Query Language
* Querying Data using the SQL Function
* SQL Insert SQL Update
* The SET command in SQL
* In-Queue SQL Statements
* In-Database Views (SQL Views)

## Problems Without Normalization

When a table is not normalised and contains redundant data, it not only takes up more space in memory but also makes it more challenging to manage and update the database without risking the loss of data. When a database is not normalised, insertion, update, and deletion anomalies (IUDs) occur often. Let's look at an actual Student table to see how these outliers work.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **rollno** | **name** | **branch** | **hod** | **office\_tel** |
| 401 | Akon | CSE | Mr. X | 53337 |
| 402 | Bkon | CSE | Mr. X | 53337 |
| 403 | Ckon | CSE | Mr. X | 53337 |
| 404 | Dkon | CSE | Mr. X | 53337 |

The data in the table above comes from four Computer Science majors. Data redundancy manifests itself in the form of field information (Head of Department) being repeated for students enrolled in the same college department.

## Normalization Rule

* There are many distinct normal forms that govern normalisation:
* Normal Form, Initial
* Forma normale secunda
* Normality, Third Form
* BCNF
* Forma Normale Quatrième

### First Normal Form (1NF)

If a table is to be considered in First Normal Form, it must adhere to the following four guidelines.

Only attributes and columns with a single value (an atom) should be included.

Column values should all belong to the same domain.

Each table column must have a distinct name.

What's more, it's irrelevant how the information is kept.

There will be extensive coverage of the First Normal Form in the subsequent lesson..

### Second Normal Form (2NF)

It is required that a table be in the First Normal Form for it to be in the Second Normal Form.

Plus, there must be no Partial Dependency.

Jump to the Second Normal Form lesson to learn about Partial Dependency and how to normalise a table to 2nd normal form.

### Third Normal Form (3NF)

Tables in second normal form are considered to be in third normal form.

Additionally, no Transitive Dependence is present.

Nonetheless, mastery of the second normal form is required before going on to the third. The Boyce-Codd Normative Form (BCNF)

As a more sophisticated extension of the Third Normal Form, the Boyce-Codd Normal Form is an increasingly useful tool in the study of complex systems. In contrast to 3NF, this form accounts for certain kinds of abnormalities. In order to be considered BCNF, a table must have a 3NF representation without any overlapping candidate keys. For a table to be considered BCNF, it must satisfy the following criteria:

Each (X Y) functional dependency should have a super Key for X and a 3rd Normal Form for R.

Using a very simple example, the Boye-Codd Normal Form course introduces the concept of BCNF and how it may be used.

### Fourth Normal Form (4NF)

Fourth Normal Form refers to the Boyce-Codd normal form of a table.

Not only that, but it lacks Multi-Valued Dependency as well.

See this page for an explanation of the Fourth Normal Form. However, you should learn the first three normal forms first before moving on to the fourth normal form.

# Why do we need a Relational Database Model?

A database management system (DBMS) is a piece of software that gives the user authority over the database itself, including its creation, definition, management, and control. The relational database model is widely utilised in many different applications as an extension of the database sitting under the relational values. Keeping financial records, logistical information, product information, employee information, and so on in a database is a standard practise. Due to its user-friendly design and management tools, it has largely supplanted older database structures, such as hierarchical and network databases. After some early setbacks, RDBMS has become a big hit and is used in many trustworthy applications because to developments like horizontal scalability of clusters and NoSQL database. In order to facilitate quick and simple retrieval of information, data points in a relational database are arranged in groups according to predetermined connections. In the relational database paradigm, the logical data structure is abstracted from the underlying physical storage, thus changes to the latter won't impact the former. Relational databases are widely utilised in business settings, since they help companies better manage their data and spot patterns in their information. Because of how easy they make it to organise and search for data, they also help businesses save time and money. When dealing with structured data, they excel.

## 4.1. What is the Concept of Relational Database?

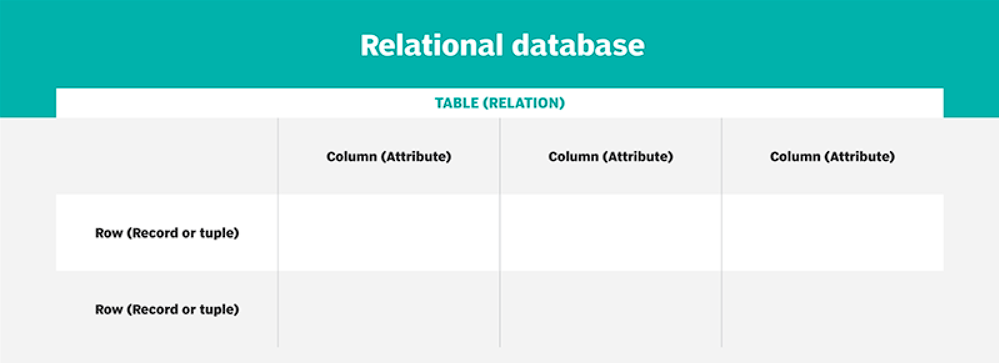
A tuple is a set of two or more items that are related to one another by some property of the object being compared. The term "tuple" is often used to describe a specific piece of data or description about an item. The tuple is said to be unique since it contains no duplicate information. Mostly Tuples may be located with the use of operations like select, and the objects themselves can be either tangible or abstract concepts. It utilises the project command to locate characteristics and the join command to merge connections. It is possible to change the connection between databases by utilising the insert, update, and delete operators. When creating a new tuple, you have the option of providing explicit values or extracting them from a query. Since the attribute of a tuple is unique and comprised of a super key, there is no need to declare it as such if any tuples have a primary key. A foreign key is a secondary column in a table that corresponds to the main column in a cross-reference relationship.

## 4.2. Relational Database Work

Information regarding linked items may be stored in the same table in a relational database. Each entry in the table has its own key, and each column has a different set of data characteristics. A value is assigned to each characteristic in each record, making it easy to see patterns in the data. Standard user and application programming interface (API) of a relational database is the Structured Query Language. Statements written in the SQL programming language are used for both ad hoc queries against a relational database and for gathering information to be utilised in reports. Data integrity standards must be followed to guarantee the reliability and availability of the relational database.

## 4.3. Relational Database Model's Structure

The relational database was created by a young IBM programmer named E. F. Codd in the year 1970. In his work "A Relational Model of Data for Large Shared Data Banks," Codd recommended moving away from keeping information in hierarchical or navigational structures and instead organising it in tables with rows and columns. Each table, or relation, in a relational database stores related pieces of information in columns called attributes. Each "row" (sometimes called a "record" or "tuple") represents a single, distinct piece of information that corresponds to one of the "keys" specified by the columns. The main key is a unique identifier for each table's data. When two tables are linked, the connection between them may be specified by using foreign keys.



A customer's name, address, phone number, and so on would all be recorded in a single table in a normal company order entry database. Details about an order, such as the item, client, date, and price, would be included in a separate table. A user may pull up a report from the database that has the specific data they need. Consider the case of a branch office manager who needs a report on all clients who made a transaction after a given date. Using the same data, a financial services manager inside the same organisation may compile a summary of overdue invoices. In a relational database, the range of values for a given data column and any associated restrictions are specified by the users. For instance, if 10 distinct clients fall under a certain domain, only three of them may be included in any given table.

# References:

*About the Sisense Application Database* (no date) *Sisense Documentation*. Available at: https://documentation.sisense.com/docs/about-the-sisense-application-database (Accessed: October 12, 2022).

By: IBM Cloud Education (no date) *Relational-databases*, *IBM*. Available at: https://www.ibm.com/cloud/learn/relational-databases (Accessed: October 12, 2022).

Lutkevich, B. and Biscobing, J. (2021) *What is a relational database?*, *SearchDataManagement*. TechTarget. Available at: https://www.techtarget.com/searchdatamanagement/definition/relational-database (Accessed: October 12, 2022).

*Relational database advantages: 8 advantages of Relational database* (2021) *EDUCBA*. Available at: https://www.educba.com/relational-database-advantages/ (Accessed: October 12, 2022).

*Sisense basic concepts and terminology* (no date) *Sisense Documentation*. Available at: https://documentation.sisense.com/docs/sisense-basic-concepts-and-terminology (Accessed: October 12, 2022).

*What is a relational database (RDBMS)?  |  google cloud* (no date) *Google*. Google. Available at: https://cloud.google.com/learn/what-is-a-relational-database (Accessed: October 12, 2022).

*What is a relational database? features & uses* (no date) *Salesforce UK Blog*. Available at: https://www.salesforce.com/uk/blog/2022/01/what-is-a-relational-database.html (Accessed: October 12, 2022).