## Project Organization

1. Reading and understanding of the project (General).

2. Create the files that will be used in the project (Text and presentation document in google drive).

3. Jose adds the group organization to the text document (Will be used in the presentation later).

4. The team divided the questionnaire equally and worked on the presentation document.

5. With the questionnaire solved, Prattya and Jose add it in the text document.

6. Alfonso is responsible for adding the organization and presentation style.

## What we learned in this project

1. Database: It's an organized collection of data, generally stored and accessed electronically from a computer system.

2. DataBase Management System (DBMS): software that interacts with end users, applications, and the database itself to capture and analyze the data.

3. DataBase uses:

    a. To support internal operations of organizations and to underpin online interactions with customers and suppliers.

    b. To hold administrative information and more specialized data, such as engineering data or economic models.

4. Relational database and a non-relational database:

| Relational database | Non-relational database |
|---|---|
| Represent and store data in tables and rows. | Represent data in collections of JSON documents. |
| They're based on a branch of algebraic set theory known as relational algebra. | Since not based on any relational theory, could be time consuming to relate one table to another |
| Relational databases use Structured Querying Language (SQL) | NoSQL and not at risk of SQL injection attacks because they are schema less |
| Allows you to link information from different tables through the use of foreign keys (or indexes) | No index linking each table |
| Lot of complicated querying, use relational databases | Databases for huge and complex datasets |
| Are not used in conjunction with Object oriented programing languages like Ruby | Are used very easily with OOPs |

Cases of relational and non relational:

| Relational database | Non-relational database |
|---|---|
| They are also called SQL Databases. | Also called NoSQL Databases. |
| The most popular of these are Microsoft SQL Server, Oracle DataBase, MYSQL and IBM DB2. | The most popular are MONGODB, DocumentDB, Cassandra, CoachBase, HBase, Redis and Neo4j. |

5. Local Database Advantages:

    a. You do not need to use concurrent access controls, data transmission, etc.

    b. it's cheaper.

    c. it requires the use of interconnected systems between several devices, since the central unit unifies everything.

6. Relational Databases Design phases:

| | |
|---|---|
| Phase 1: Collect user requirements. | Identify needs. |
| Phase 2: Conceptual design. | We will shape our entities and the relationships that will exist between them. |
| Phase 3: Logical design. | In this phase, we must think about how to normalize our tables to avoid duplication of information and to save storage space. |
| Phase 4: Physical design. | we will shape our entities and the relationships that will exist between them. |

7. Some SGBD:

    a. Oracle database. Object-relational database management system developed by Oracle Corporation. It is considered one of the most complete database systems and until recently it had a great market dominance.

    b. Microsoft SQL Server Database management system of the relational model developed by Microsoft. It is only available for the Windows operating system.

    c. PostgreSQL. Object-oriented database management system. It is open source and is published under a BSD license.

    d. MariaDB. A derivation of MySQL.

    e. MongoDB. We are facing the Non-relational Database Management System (DBMS NoSQL).

8. Database inconsistency: This occurs when data is unnecessarily repeated in the files or tables that make up the database. We say that there is data redundancy when the same information is stored several times in the same system or tables. The duplicated data, also known as the redundant data, creates unreliable information because the chances of having a value changed in one file are high, but on the other file the value remains the same. This condition of inconsistency is often experienced when using the traditional file processing, and it is very expensive and difficult to rectify such inconsistencies.

9. Master Table: The master tables in all relational databases are those that contain information that does not change over time. The design of the master tables i.e. its columns and constraints describe the entities in the system. Its purpose is to capture the system.

10. Static and Dynamic DataBase:

| Static DataBase | Dynamic DataBase |
|---|---|
| Static or Embedded SQL are SQL statements in an application that do not change at runtime. | It's a programming technique that allows you to build SQL Statements dynamically at runtime. |
| For example, the application may allow users to enter their own queries. | You can create a more general purpose, flexible applications by using dynamic SQL because the full text of a SQL statement may be unknown at compilation. |

11. The SQL Language:It's a domain-specific language designed for managing data.

- SQL Data can be held in:

    a. A relational database management system (RDBMS).

    b. Stream processing in a relational data stream management system (RDSMS).

- Useful in handling structured data (data incorporating relations among entities and variables).

12. The list of main database queries could be explained as:

a. **Select Query:** The select query is the simplest type of query and because of that, it is also the most commonly used one in Microsoft Access databases. It can be used to select and display data from either one table or a series of them depending on what is needed. In the end, it is the user-determined criteria that tells the database what the selection is to be based on. After the select query is called, it creates a "virtual" table where the data can be changed, but at no more than one record at a time.

b. **Action Query:** When the action query is called, the database undergoes a specific action depending on what was specified in the query itself. This can include such things as creating new tables, deleting rows from existing ones and updating records or creating entirely new ones. Four kinds of action queries are:
    i. Append Query– takes the set results of a query and "appends" (or adds) them to an existing table.
    ii. Delete Query– deletes all records in an underlying table from the set results of a query.
    iii. Create Query– as the name suggests, it creates a table based on the set-results of a query.
    iv. Update Query– allows for one or more field in your table to be updated.

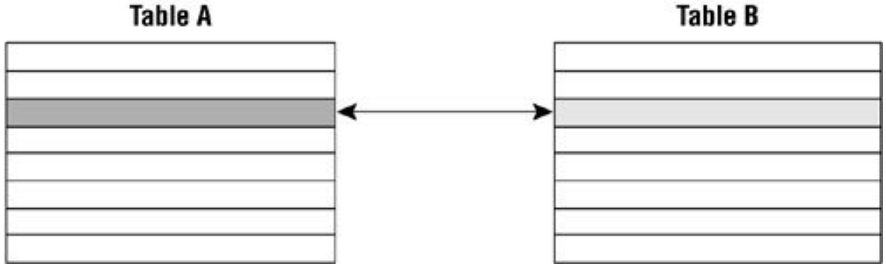
c. **Parameter Query:** A parameter query works with other types of queries to get whatever results you are after. This is because, when using this type of query, you are able to pass a parameter to a different query, such as an action or a select query. It can either be a value or a condition and will essentially tell the other query specifically what you want it to do. It is often chosen because it allows for a dialog box where the end user can enter whatever parameter value they wish each time the query is run. The parameter query is just a modified select query.

d. **Aggregate Query:** A special type of query is known as an aggregate Query. It can work on other queries (such as selection, action or parameter) just like the parameter query does, but instead of passing a parameter to another query it totals up the items by selected groups. It essentially creates a summation of any selected attribute in your table. This can be further generated into statistical amounts such as averages and standard deviation, just to name a couple.

13. Physical Data-model: A physical data model defines all of the logical database components and services that are required to build a database or can be the layout of an existing database. A physical data model consists of the table's structure, column names and values, foreign and primary keys and the relationships among the tables. A physical data model primarily defines all the relational data models and objects of the database. It is created using the native database language of the database management system (DBMS). It can also be created by transforming the logical model. A physical data model is used by database administrators to estimate the size of the database systems and to perform capacity planning. The physical data model constraints such as size, configuration and security can vary based on the underlying database system.

14. Types of database relationships:

- One to one relationships: A pair of tables bears a one-to-one relationship when a single record in the first table is related to only one record in the second table, and a single record in the second table is related to only one record in the first table.




As you can see, a single record in TABLE A is related to only one record in TABLE B, and a single record in TABLE B is related to only one record in TABLE A. A one-to-one relationship usually (but not always) involves a subset table.

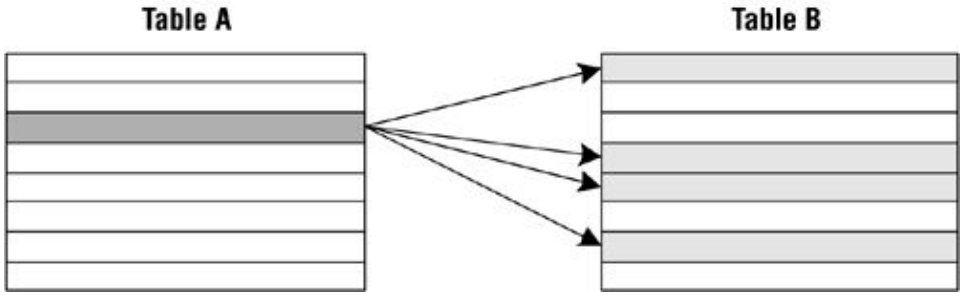
**Employees**

| EmpID | EmpFirst Name | EmpLast Name | Home Phone | *<< other fields >>* |
|---|---|---|---|---|
| 100 | Zachary | Erlich | 553-3992 | ..... |
| 101 | Susan | McLain | 790-3992 | ..... |
| 102 | Joe | Rosales | 551-4993 | ..... |

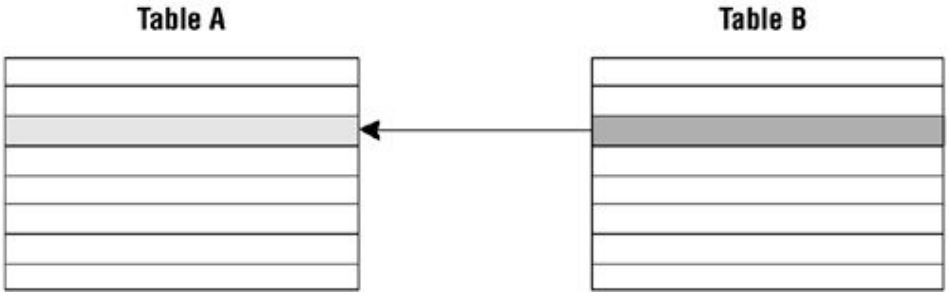
**Compensation**

| EmpID | Hourly Rate | Commission Rate | *<< other fields >>* |
|---|---|---|---|
| 100 | 25.00 | 5.0% | ..... |
| 101 | 19.75 | 3.5% | ..... |
| 102 | 22.50 | 5.0% | ..... |

- One to many relationships: one-to-many relationship exists between a pair of tables when a single record in the first table can be related to one or more records in the second table, but a single record in the second table can be related to only one record in the first table.Because of the relationship, a single record in TABLE A can be related to one or more records in TABLE B.




Conversely, a single record in the TABLE B can be related to only one record in TABLE A.



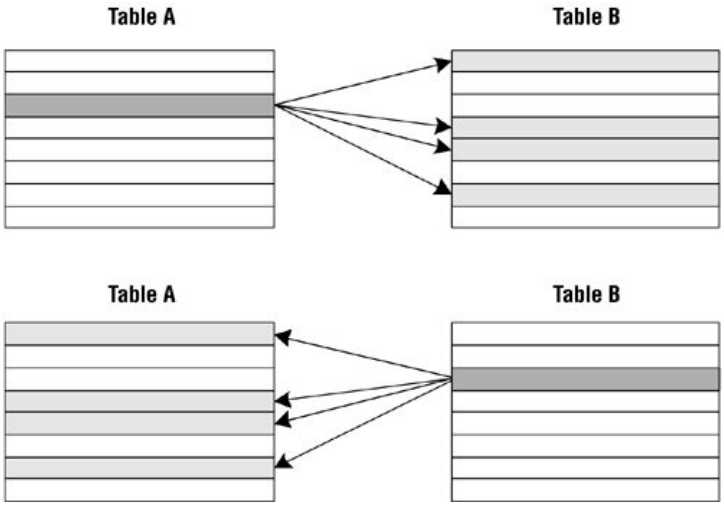

**Customers**

| Customer ID | CustFirst Name | CustLast Name | << other fields >> |
|---|---|---|---|
| 9001 | Paul | Litwin | ..... |
| 9002 | Alison | Balter | ..... |
| 9003 | Andy | Baron | ..... |
| 9004 | Chris | Kunicki | ..... |
| 9005 | Mary | Chipman | ..... |

**Customer Rentals**

| Customer ID | Video ID | Checkout Date |
|---|---|---|
| 9002 | 80115 | 09/26/01 |
| 9001 | 64558 | 09/28/01 |
| 9003 | 10202 | 09/28/01 |
| 9003 | 11354 | 09/28/01 |
| 9003 | 78422 | 10/02/01 |
| 9005 | 30556 | 09/26/01 |
| 9004 | 20655 | 10/05/01 |

- Many to many relationships: A pair of tables bears a many-to-many relationships when a single record in the first table can be related to one or more records in the second table and a single record in the second table can be related to one or more records in the first table. Assume once again that you're working with TABLE A and TABLE B and that there is a many-to-many relationships between them. Because of the relationship, a single record in TABLE A can be related to one or more records (but not necessarily all) in TABLE B. Conversely, a single record in the TABLE B can be related to one or more records (but not necessarily all) in TABLE A.




This is the second most common relationship that exists between a pair of tables in a database. It can be a little more difficult to identify than a one-to-many relationship, so you must be sure to examine the tables carefully.

**Students**

| Student ID | StudFirst Name | StudLast Name | StudStreet Address | StudCity | StudState | StudZipcode | *<< other fields >>* |
|---|---|---|---|---|---|---|---|
| 60001 | Zachary | Erlich | 1204 Bryant Road | Seattle | WA | 98125 | ..... |
| 60002 | Susan | McLain | 101 C Street, Apt. 32 | Redmond | WA | 98052 | ..... |
| 60003 | Joe | Rosales | 201 Cherry Lane SE | Redmond | WA | 98073 | ..... |
| 60004 | Diana | Barlet | 4141 Lake City Way | Woodinville | WA | 98072 | ..... |
| 60005 | Tom | Wickerath | 2100 Mineola Avenue | Bellevue | WA | 98006 | ..... |

**Classes**

| Class ID | Class Name | Class Category | Credits | Instructor ID | Classroom | *<< other fields >>* |
|---|---|---|---|---|---|---|
| 900001 | Advanced Calculus | Math | 5 | 220087 | 2201 | ..... |
| 900002 | Advanced Music Theory | Music | 3 | 220039 | 7012 | ..... |
| 900003 | American History | History | 5 | 220148 | 3305 | ..... |
| 900004 | Computers in Business | Computer Science | 2 | 220387 | 5115 | ..... |
| 900005 | Computers in Society | Computer Science | 2 | 220387 | 5117 | ..... |
| 900006 | Introduction to Biology | Biology | 5 | 220498 | 3112 | ..... |
| 900007 | Introduction to Database Design | Computer Science | 5 | 220516 | 5105 | ..... |
| 900008 | Introduction to Physics | Physics | 4 | 220087 | 2205 | ..... |
| 900009 | Introduction to Political Science | Political Science | 5 | 220337 | 3308 | ..... |

A student can attend one or more classes during a school year, so a single record in the STUDENTS table can be related to one or more records in the CLASSES table. Conversely, one or more students will attend a given class, so a single record in the CLASSES table can be related to one or more records in the STUDENTS table. The main problems with many to many databases are:

I. It will be tedious and somewhat difficult to retrieve information from one of the tables.
II. One of the tables will contain a large amount of redundant data.
III. Duplicate data will exist within both tables.
IV. It will be difficult for you to insert, update, and delete data.

15. Databases that can be stored in main data engines:

357 systems in r

| Rank | | | DBMS | Database Model |
|---|---|---|---|---|
| Nov 2019 | Oct 2019 | Nov 2018 | | |
| 1. | 1. | 1. | Oracle | Relational, Multi-model |
| 2. | 2. | 2. | MySQL | Relational, Multi-model |
| 3. | 3. | 3. | Microsoft SQL Server | Relational, Multi-model |
| 4. | 4. | 4. | PostgreSQL | Relational, Multi-model |
| 5. | 5. | 5. | MongoDB | Document, Multi-model |
| 6. | 6. | 6. | IBM Db2 | Relational, Multi-model |
| 7. | 7. | ↑ 8. | Elasticsearch | Search engine, Multi-model |
| 8. | 8. | ↓ 7. | Redis | Key-value, Multi-model |
| 9. | 9. | 9. | Microsoft Access | Relational |
| 10. | 10. | ↑ 11. | Cassandra | Wide column |

- Oracle:- Relational, Document store, Graph DBMS, RDF store.
- MySQL:- Relational and Document store.
- Microsoft SQL server:- Relational, Document store, Graph DBMS.
- PostgreSQL:- Relational and Document store.
- MongoDB:- Document store and search engine.

16. Primary Key: A special relational database column or columns to uniquely identify all table records.

| Important characteristics |
| --- |
| Cannot be null |
| Must be unique value for each row |

| A good primary key must have the following attributes |
| --- |
| 1. Keep it short |
| 2. Use a number |
| 3. Keep it simple |
| 4. Never change |

17. Auto-increment: Auto increment allows for the generation of a unique key once a new row is inserted.

18.

| The not_null | Constraints the column not to accept NULL values. |
| --- | --- |
| Unique | Constraints all values in the column to be different |
| Index | Used to create index in tables |

19. Graphical tools:

- RazorSQL
- Microsoft SQL Server Management Studio
- MySQL Workbench
- TeamDesk
- TablePlus
- Sequel Pro
- phpMyAdmin
- Navicat for MySQL
- SQLyog
- Knack