Relational Databases Anomalies Normalization

.NET CORE

*A Relational Database is made up of a collection of **tables** that stores a specific set of structured data. A table contains a collection of rows (**tuples**) and columns (**attributes**). Each column in the table is designed to store a certain type of data.*

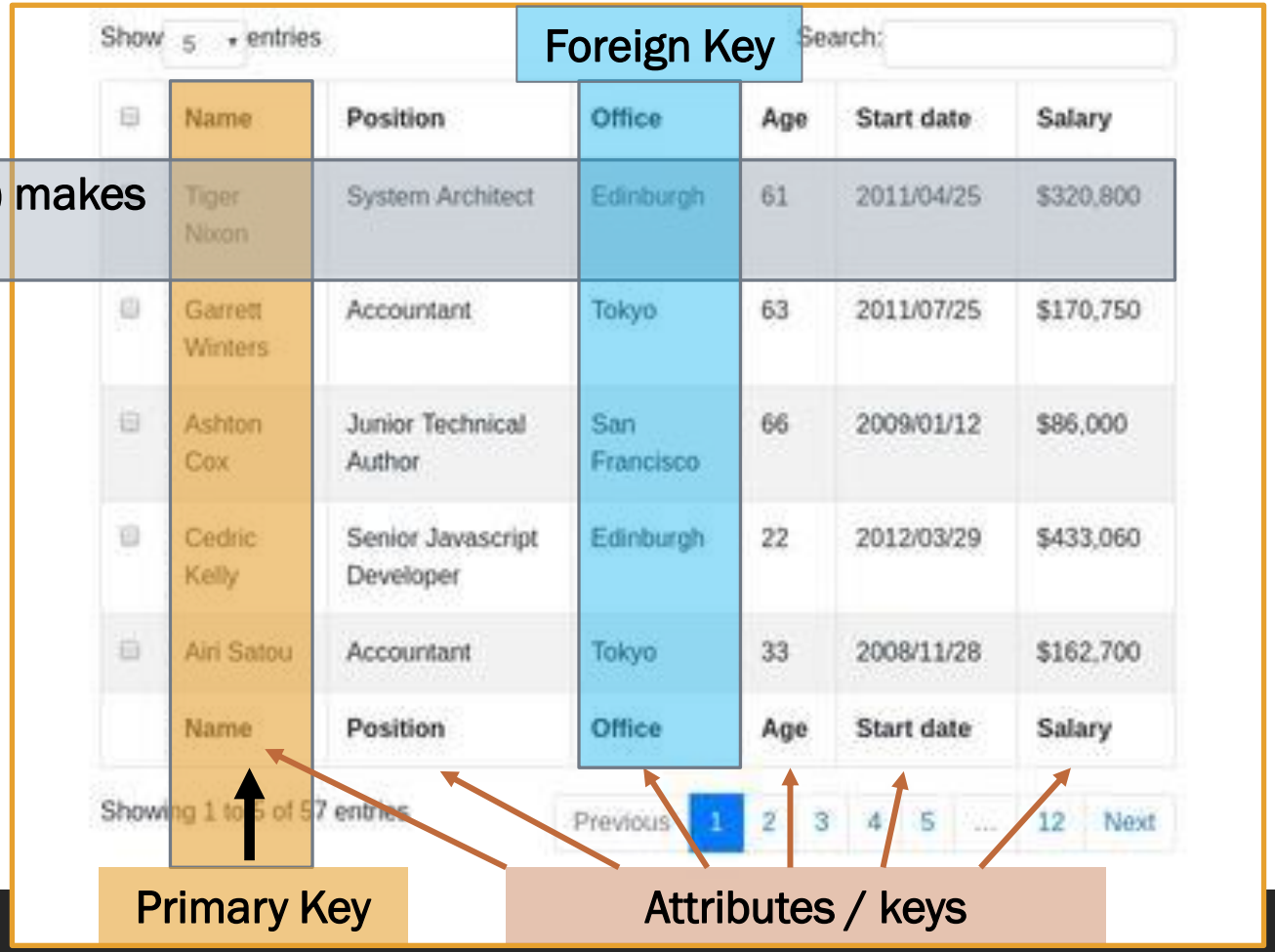
https://docs.microsoft.com/en-us/sql/relational-databases/databases/databases?view=sql-server-ver15

Databases – Instances/Users

https://docs.microsoft.com/en-us/sql/relational-databases/databases/databases?view=sql-server-ver15

There are one or more ***schemas*** within a database. Within each **schema** there are database objects such as ***tables*** and ***views***.

This is a ***table***. →→→→→



DBMS (Database Management System)

https://www.tutorialspoint.com/dbms/dbms_overview.htm

- Data is a collection of facts and figures that can be processed to produce information.
- A Database is a collection of related data.
- A ***DBMS*** stores data in such a way that it becomes easier to retrieve and manipulate.

If we have data about grades of all students, we can extrapolate average grades and highest grades.

Primary Key

Last Name	Birthday	Grade 1	Grade 2	Grade 3	Final Grade
Smith	June 7, 1987	100	98	89	95.67
Jones	October 5, 1986	75	89	84	82.67
Garcia	December 15, 1986	99	97	100	98.67
Kim	February 28, 1987	50	68	42	53.33
Washington	May 4, 1987	85	87	79	83.67
Hernandez	October 8, 1986	74	72	81	75.67
Gates	March 21, 1987	32	54	67	51.00
Papert	April 26, 1987	84	92	81	85.67
Kennedy-Onassis	January 18, 1987	92	90	88	90.00
Smith	February 8, 1987	72	65	99	78.67

DBMS - Benefits

https://www.tutorialspoint.com/dbms/dbms_overview.htm

Relation-based tables – A Relational DBMS allows ***entities*** and ***relations*** among them to form ***tables***.

Separation of data and application – A database is separated from its data. A database is an active entity, whereas data on which the database works is said to be passive.

Less redundancy – DBMS follows the rules of ***normalization***, which splits a relation when any of its attributes has redundancy.

Consistency – Consistency is the state where every relation in a database is ***persisted***.

Query Language – DBMS (or RDBMS) is equipped with query language (usually ***SQL***), which makes it more efficient to retrieve and manipulate data.

ACID Properties – A DBMS follows the concepts of ***Atomicity***, ***Consistency***, ***Isolation***, and ***Durability***.

Isolation Levels – A DBMS supports a multi-user environment and applies restrictions on the access and manipulation of data in parallel.

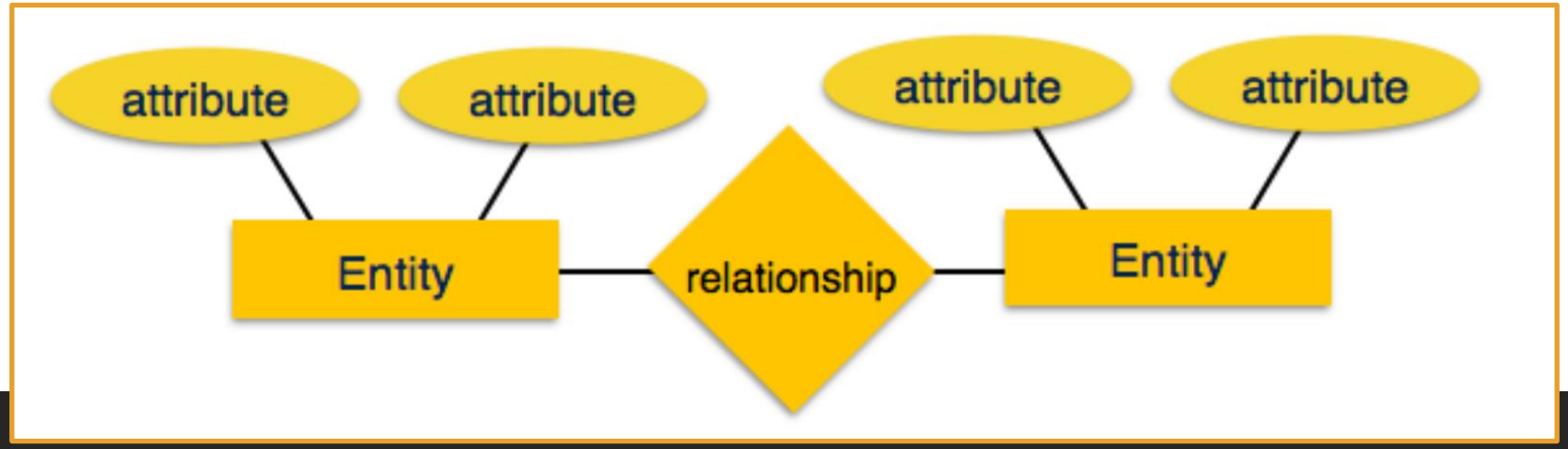
Database – Entity-Relationship Models

https://www.tutorialspoint.com/dbms/dbms_data_models.htm

Data models define the structure of a database. ***Data Models*** are ***entities*** that introduce abstraction in a ***DBMS***.

Data models define how data is connected to each other and how they are processed and stored inside the database.

An ***Entity-Relationship (ER) Model*** is based on the notion of real-world entities and the relationships between them. An ***ER Model*** is used for the conceptual design of a database.



Database – Schema Diagram

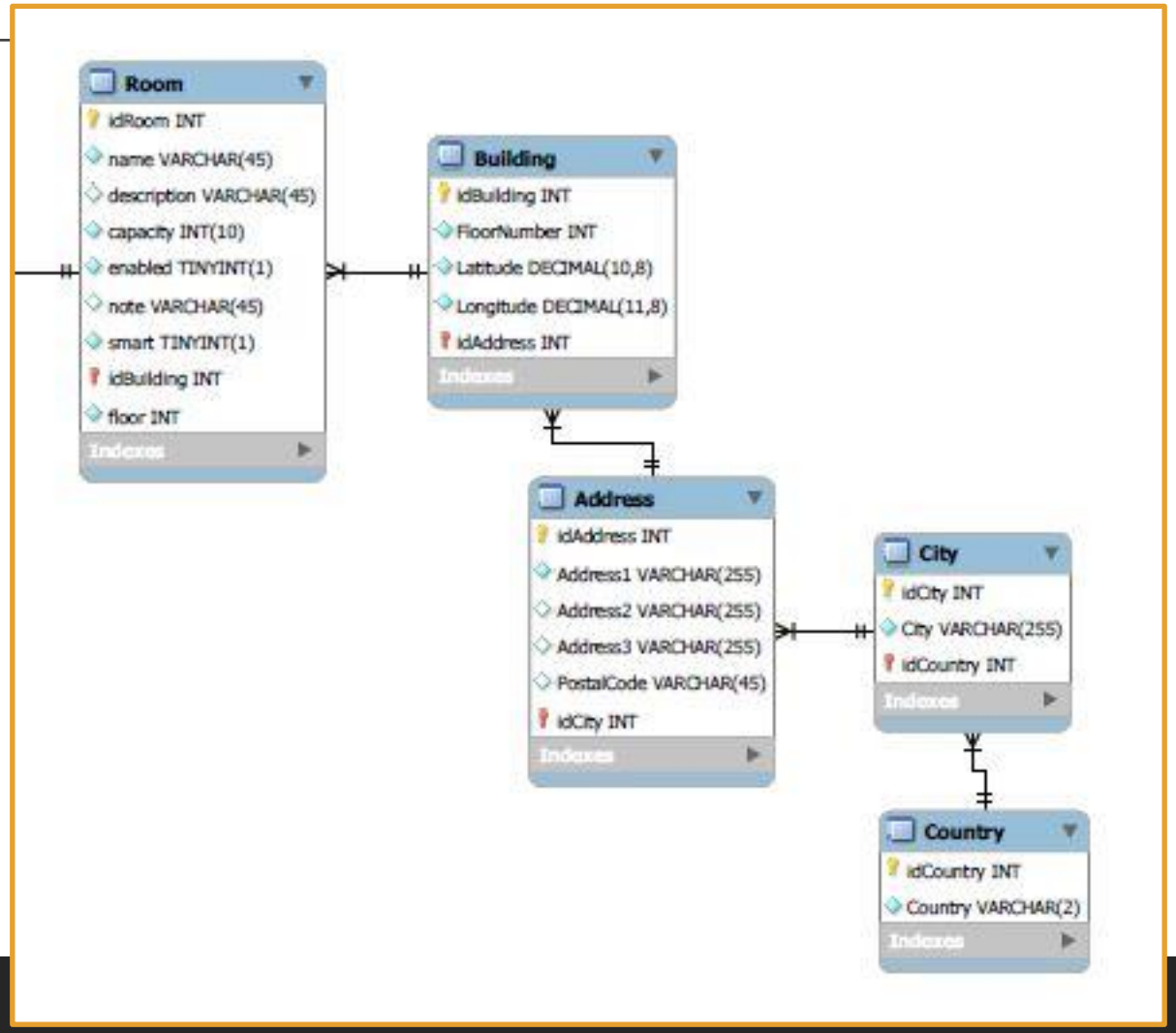
https://www.tutorialspoint.com/dbms/dbms_data_schemas.htm

A database ***schema*** is the skeleton structure that represents the logical view of the entire database.

It defines how the data is organized and how the ***relations*** among them are associated.

It displays all the ***constraints*** that are to be applied on the data.

A ***schema diagram*** contains a descriptive detail of the database.

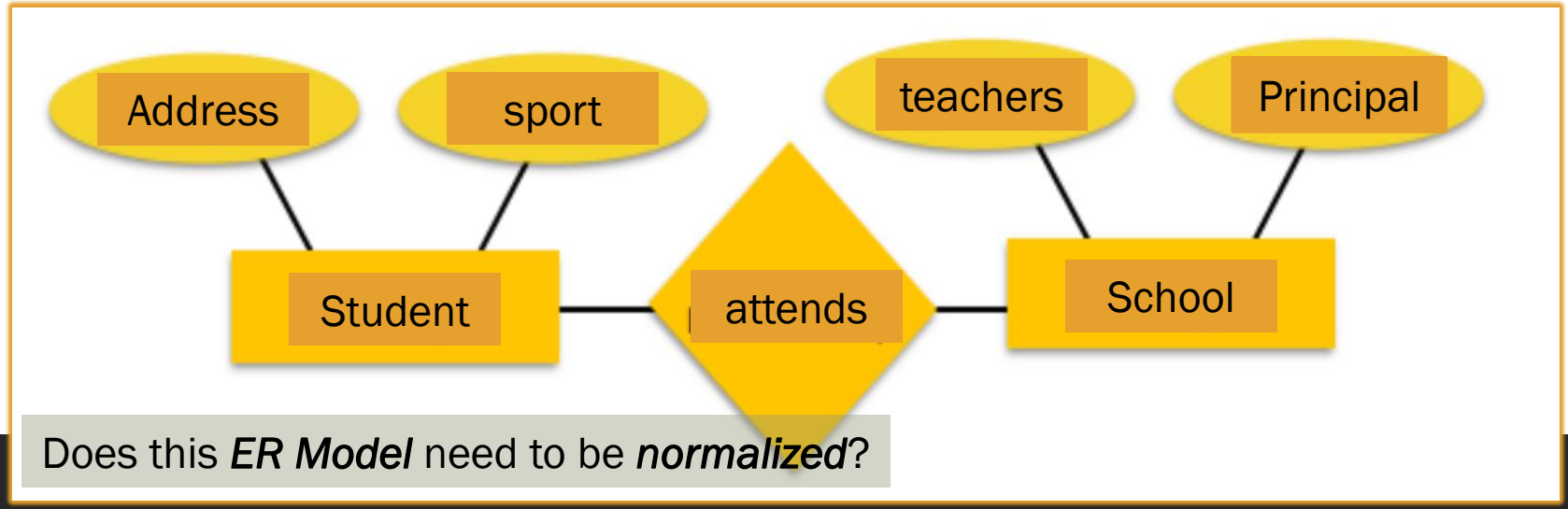


Database – Entity-Relationship Models

https://www.tutorialspoint.com/dbms/dbms_data_models.htm

- **Entity** – a real-world thing having properties called **attributes**.
- **Attribute** - the details about an ***entity***.
- **Relationship** – The logical association among entities. Relationships are mapped with entities in various ways.
- "Mapping cardinalities" defines the number of associations between two entities.

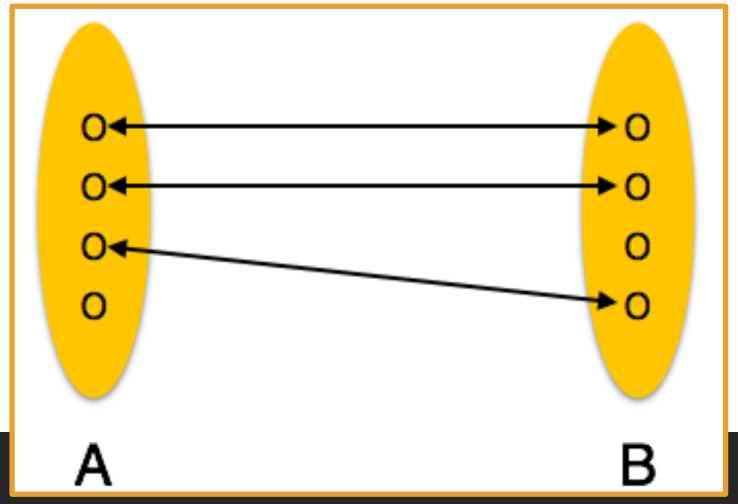
- Mapping cardinalities –
  - *one to one*
  - *one to many*
  - *many to many*

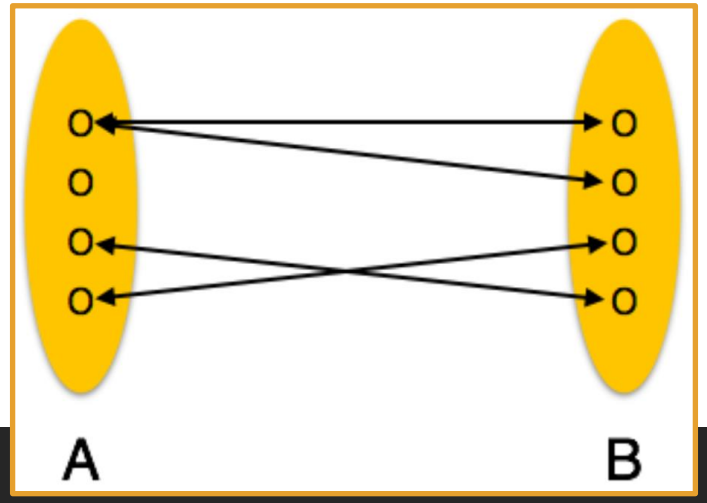


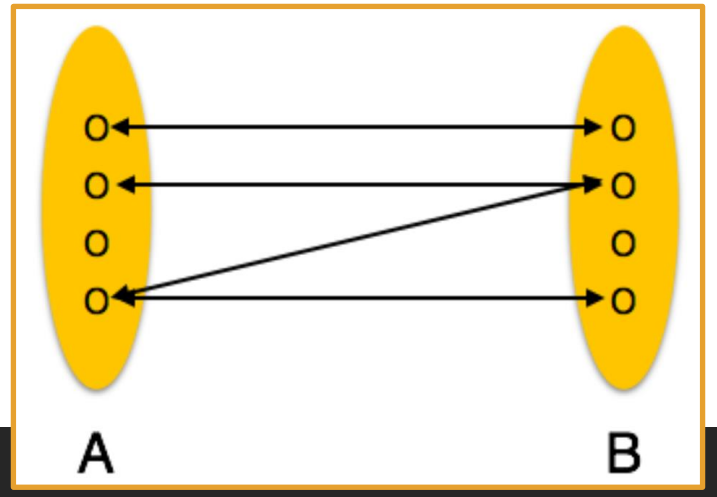
Database – Relationships

https://www.tutorialspoint.com/dbms/er_model_basic_concepts.htm

Cardinality		
One-to-one	One-to-many	Many-to-many
One ***entity*** from ***entity*** set A can be associated with at most one ***entity*** of ***entity*** set B and vice versa.	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 other ***entity***	One ***entity*** from A can be associated with more than one ***entity*** from B and vice versa.







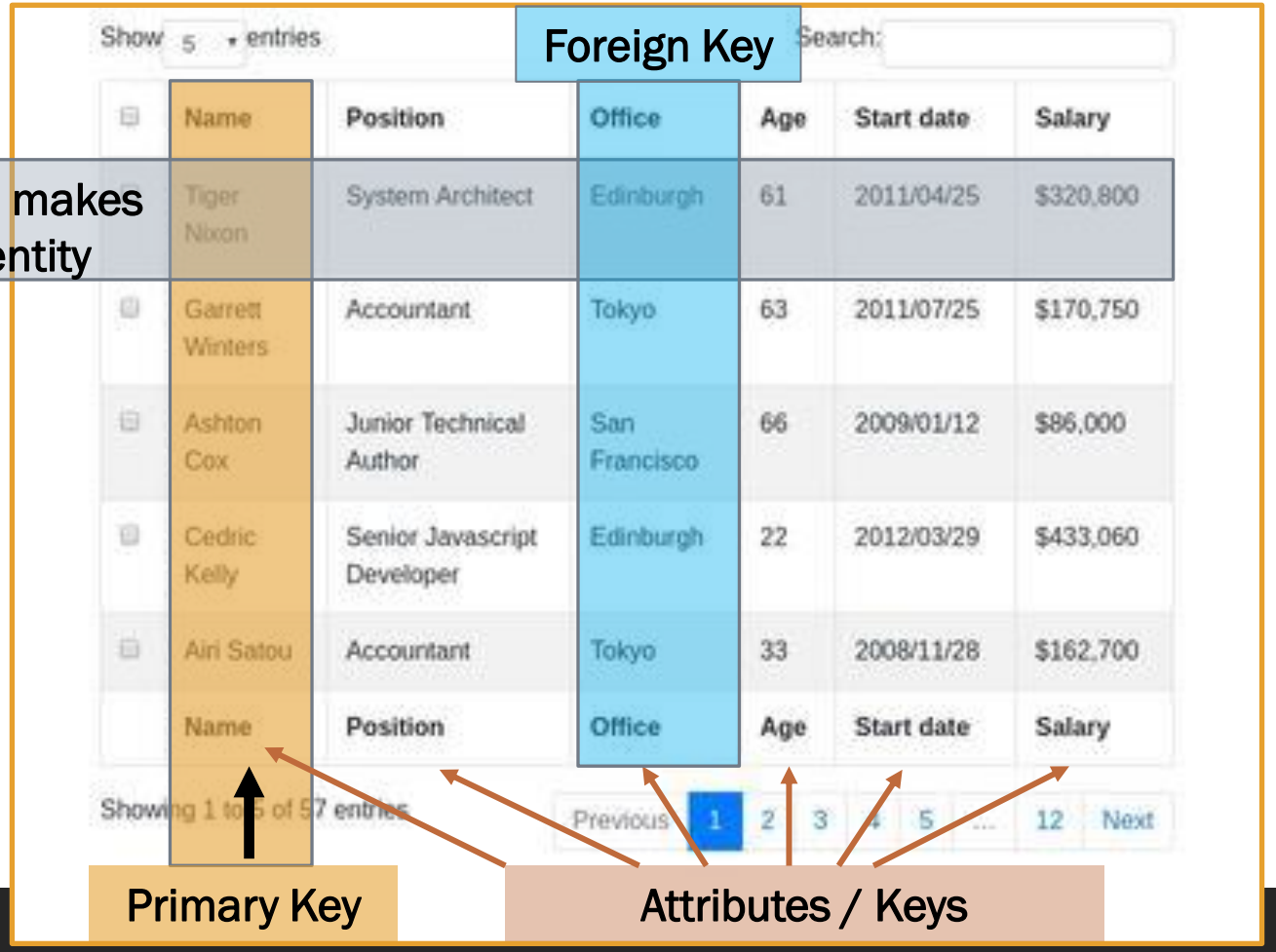
Databases – Primary and Candidate Keys

https://www.tutorialspoint.com/dbms/er_model_basic_concepts.htm
https://docs.microsoft.com/en-us/ef/core/modeling/keys?tabs=data-annotations

Candidate Key – An ***attribute*** of an entity. An ***entity*** set may have more than one ***candidate key***.

Primary Key – A ***Candidate Key*** chosen to uniquely identify the ***entity*** set(***tuple***).

Foreign Key – The ***Primary Key*** of another table. Used to represent to other entity.



Database - Keys in Entity Framework

https://docs.microsoft.com/en-us/ef/core/modeling/keys?tabs=data-annotations

A ***key*** serves as a unique identifier for each ***entity*** instance. Most ***entities*** in ***EF*** have a single ***key***, which maps to the concept of a ***primary key*** in relational databases. It's possible for an ***entity*** to have no keys. ***Entities*** can have additional ***keys*** (Alternate ***Keys***) beyond the ***Primary Key***. By convention, any property named ***Id*** or [type name]Id will be automatically configured by ***EF*** as the ***Primary Key*** of an ***entity***.

You can force configure any single property to be the Primary Key of an entity.

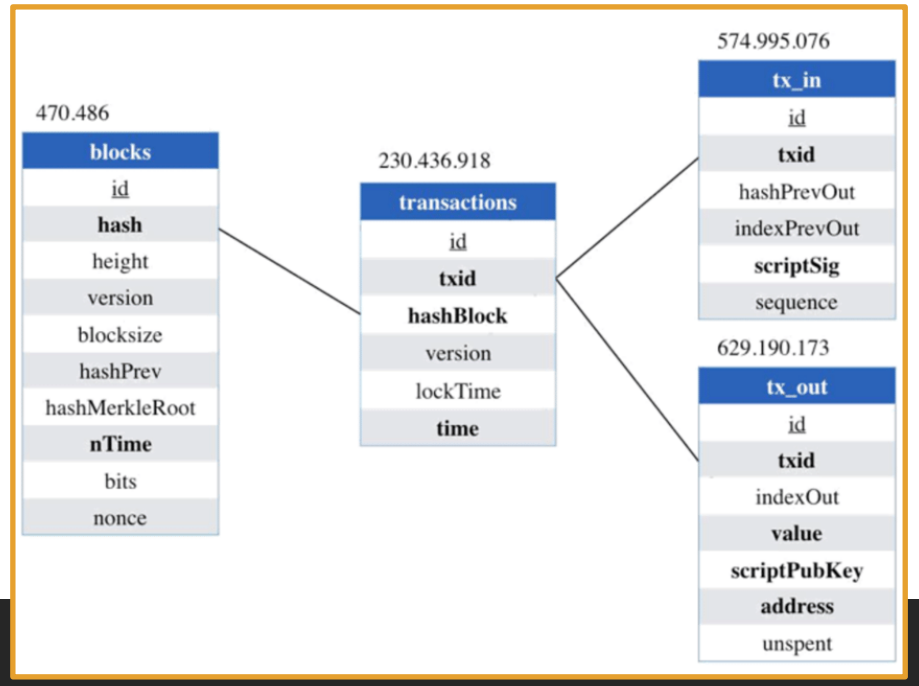
```
class Car
{
    [Key]
    public string LicensePlate { get; set; }

    public string Make { get; set; }
    public string Model { get; set; }
}
```

Relational Databases

https://en.wikipedia.org/wiki/Relational_database#RDBMS

A ***RDBMS*** allows ***entities*** to form ***tables*** with ***relations*** between them. You could understand the architecture of a database just by looking at the ***table*** names.



Students : Table

	StudentID	LastName	FirstName	State	Zip
+	151	Galla	Kristi	NJ	19038
+	152	McLaughlin	Liz	NY	19001
+	153	Greber	Christopher	DE	19446
+	154	Flowers	Lucy	DE	19002
+	155	Joranne	Scott	PA	19422
+	157	Kents	Brian	NY	19403
+	158	Keener	William	NJ	19444
+	159	Nguyen	Ted	DC	19401
+	160	Hauny	Mindy	DE	19426
+	161	Burcik	Reber	DE	19446
+	162	Fallows	Keith	NY	19440
+	163	Rahn	Sid	PA	19438
+	164	O'Mara	John	NY	19002
+	165	Welsh	Irma	NY	19473
+	166	Bewley	Red	NJ	19002
+	171	Edgar	Ernest	PA	19027
+	172	Legrow	Bill	DE	19415
+	173	Davenport	David	DE	19002

Record: 1 of 197

First Name	Last Name	Birthday	Grade 1	Grade 2	Grade 3	Final Grade
John	Smith	June 7, 1987	100	98	89	95.67
Kathleen	Jones	October 5, 1986	75	89	84	82.67
Juanita	Garcia	December 15, 1986	99	97	100	98.67
Charles	Kim	February 28, 1987	50	68	42	53.33
Natalie	Washington	May 4, 1987	85	87	79	83.67
James	Hernandez	October 8, 1986	74	72	81	75.67
William	Keener	March 21, 1987	32	54	67	51.00
Seymour	Papert	April 26, 1987	84	92	81	85.67
Jackie	Kennedy-Onassis	January 18, 1987	92	90	88	90.00
Timothy	Smith	February 8, 1987	72	65	99	78.67

Relational Databases – Concepts

https://www.tutorialspoint.com/dbms/relational_data_model.htm

---

Tables – "relations" are saved in table format. This format stores the relation among ***entities***. A table has rows and columns, where rows represent ***entities*** and columns represent ***attributes***.

Tuple – A single row of a ***table***, which contains a single record for that ***entity*** is called a ***tuple***.

Relation schema – A relation schema describes the relation name (table name), attributes, and their names.

Primary Key – Each row has one (or more) attributes, chosen as ***Primary keys***. These identify the row in the ***table*** uniquely.

Composite Key – Multiple ***Candidate Keys*** that together form the ***Primary Key***.

Index – A unique number given to each ***tuple*** in a ***table*** to serve as the ***Primary Key***.

Relational Databases – Constraints

https://www.tutorialspoint.com/dbms/relational_data_model.htm

Key Constraints	Domain Constraints	Referential Integrity Constraints
Candidate Keys must uniquely identify an entity. A ***Candidate Key*** can not have NULL values.	Every ***attribute*** must have a specific range of values.	A ***Foreign Key*** refers to a ***Primary key*** of a different ***table***. If a ***tuple*** has a ***Foreign Key***, that ***PK*** must exist.

Databases – Anomalies

https://www.tutorialspoint.com/dbms/database_normalization.htm

If a Database has inconsistent data, it will incur ***anomalies***. A DB with ***anomalies*** can give inconsistent data.

There are three types of anomalies:

- ***Update anomaly*** – If data items are not linked to each other properly, when one data item is updated, a few instances may get updated properly while a few others are left with old values
- ***Deletion anomaly*** – When a record is deleted, but linked parts of it were left undeleted because of unawareness or when deletion deletes other data unintentionally.
- ***Insertion anomaly*** – When data is inserted into a record that does not exist or cannot be inserted without an unrelated data.

Normalization Assignment- Create an unnormalized table.

List the information of your family members.

There must be at least 5 attributes to each tuple and at least 5 entities.

Databases – Normalization

https://www.tutorialspoint.com/dbms/database_normalization.htm
https://www.c-sharpcorner.com/UploadFile/0146e3/database-normalization/

Normalization is a method to prevent ***anomalies*** and keep the database in a consistent state. ***Fields*** and ***tables*** of a relational DB are organized to minimize redundancy and dependency.

Normalization involves dividing large ***tables*** into smaller (and less redundant) ***tables*** and defining relationships among their ***atomic*** data.

There are many normal forms but *1NF*, *2NF*, and *3NF* are primarily used.

Databases – Normalization

https://www.tutorialspoint.com/dbms/database_normalization.htm
https://www.c-sharpcorner.com/UploadFile/0146e3/database-normalization/

This table is not normalized. All the information is stored in one table.

SALUTATION	CUSTOMER NAME	CITY	BOOK ISSUED
MR.	RAJ	BANGALORE	LET US C,ORACLE DATABSE,
MISS.	PRIYA	CHENNAI	PROGRAMMING WITH JAVA,C++ PROGRAMMING
MR.	RAJ	DELHI	DBA FUNDAMENTALS, ORACLE PROGRAMMING

Databases – Normalization

https://www.tutorialspoint.com/dbms/database_normalization.htm
https://www.c-sharpcorner.com/UploadFile/0146e3/database-normalization/

1st Normal Form (1NF) –

- each table cell should contain a single value.
- Each record needs to be unique.
- It contains atomic values only.

SALUTATION	CUSTOMER NAME	CITY	Book Issued
MR.	Raj	BANGALORE	LET US C
MR.	Raj	BANGALORE	ORACLE DATABSE
Miss	Priya	CHENNAI	PROGRAMMING WITH JAVA
Miss	Priya	CHENNAI	C++ PROGRAMMING
MR.	Raj	DELHI	DBA FUNDAMENTALS
MR.	Raj	DELHI	ORACLE PROGRAMMING

Databases – Normalization

https://www.tutorialspoint.com/dbms/database_normalization.htm
https://www.c-sharpcorner.com/UploadFile/0146e3/database-normalization/

2nd Normal Form (2NF) –

- First, be in ***1NF***.
- Have a single ***Primary Key***.
- Remove subsets of data that apply to multiple rows of a ***table*** and place them in separate ***tables*** with ***PK → FK*** relationships among the new tables.
- If the table is in ***1NF*** and every non-key attribute is dependent on the ***Primary Key***. then 2NF is also achieved.

The ***1NF*** table is divided into two tables.
Table 1 contains only member information.
Membership_id is created to point to the ***Primary Key*** for table 1.
Table 2 contains the information for each book. Table 2's new column is ***book_id***. It is the ***Primary Key*** for table 2.

1

MEMBERSHIP ID	SALUTATION	CUSTOMER NAME	CITY
1	MR.	RAJ	BANGALORE
2	MISS.	PRIYA	CHENNAI
3	MR.	RAJ	DELHI

2

BOOK ID	MEMBERSHIP ID	BOOK ISSUED
1	1	LET US C
2	1	ORACLE DATABSE
3	2	PROGRAMMING WITH JAVA
4	2	C++ PROGRAMMING
5	3	ORACLE PROGRAMMING
6	3	DBA FUNDAMENTALS

Databases – Normalization

https://www.tutorialspoint.com/dbms/database_normalization.htm
https://www.c-sharpcorner.com/UploadFile/0146e3/database-normalization/

To achieve 3NF Normalization, there must be no dependencies between fields in a single row. This means: "Given a value for A, do we then have only one possible value for B, and vice versa?"

A ***transitive functional dependency*** occurs when the change of a ***Candidate Key*** column might cause any of the other ***Candidate Key*** columns to change. In table 1, changing the non-key column *Customer Name* may change *Salutation.*

1

MEMBERSHIP ID	SALUTATION	CUSTOMER NAME	CITY
1	MR.	RAJ	BANGALORE
2	MISS.	PRIYA	CHENNAI
3	MR.	RAJ	DELHI

If the answer is yes, then A and B should be put into a new table, with A becoming the ***Primary Key***. A reference to A should be left in the original relation and marked as a ***Foreign Key***.

2

BOOK ID	MEMBERSHIP ID	BOOK ISSUED
1	1	LET US C
2	1	ORACLE DATABSE
3	2	PROGRAMMING WITH JAVA
4	2	C++ PROGRAMMING
5	3	ORACLE PROGRAMMING
6	3	DBA FUNDAMENTALS

Databases – Normalization

https://www.tutorialspoint.com/dbms/database_normalization.htm
https://www.c-sharpcorner.com/UploadFile/0146e3/database-normalization/

A ***transitive functional dependency*** occurs when the change of a ***Candidate Key*** column might cause any of the other ***Candidate Key*** columns to change. In table 1, changing the non-key column *Customer Name* may change *Salutation*.

Table 1

MEMBERSHIP ID	SALUTATION ID	CUSTOMER NAME	CITY
1	1	RAJ	BANGALORE
2	2	PRIYA	CHENNAI
3	1	RAJ	DELHI

Table 3

ID	SALUTATION NAME
1	MR.
2	MISS
3	MRS.
4	DR.

Table 2

BOOK ID	MEMBERSHIP ID	BOOK ISSUED
1	1	LET US C
2	1	ORACLE DATABSE
3	2	PROGRAMMING WITH JAVA
4	2	C++ PROGRAMMING
5	3	ORACLE PROGRAMMING
6	3	DBA FUNDAMENTALS

The table is divided again, and a new table is created that stores Salutation only. The database is now in ***3NF***.

Assignment-
Convert your table to a 3NF table.

List the information of your family members.

There must be at least 5 attributes to each tuple and at least 5 entities.