Relational Db and Anomalies

.NET

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

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

Relational Database – 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***.

RDBMS - Benefits

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

Relation-based tables – A Relational DBMS allows tables to store related data in the form of ***entities*** that can be queried and combined.

Separation of the data (the state) and the application – A database is logically separated from the data it holds. A database is an active entity, whereas data on which the database works is said to be passive.

Less redundancy – DBMS follows rules of ***Normalization***. Normalization splits a relation when any of its attributes has redundancy.

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

Query Language – DBMS (or RDBMS) has a 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.

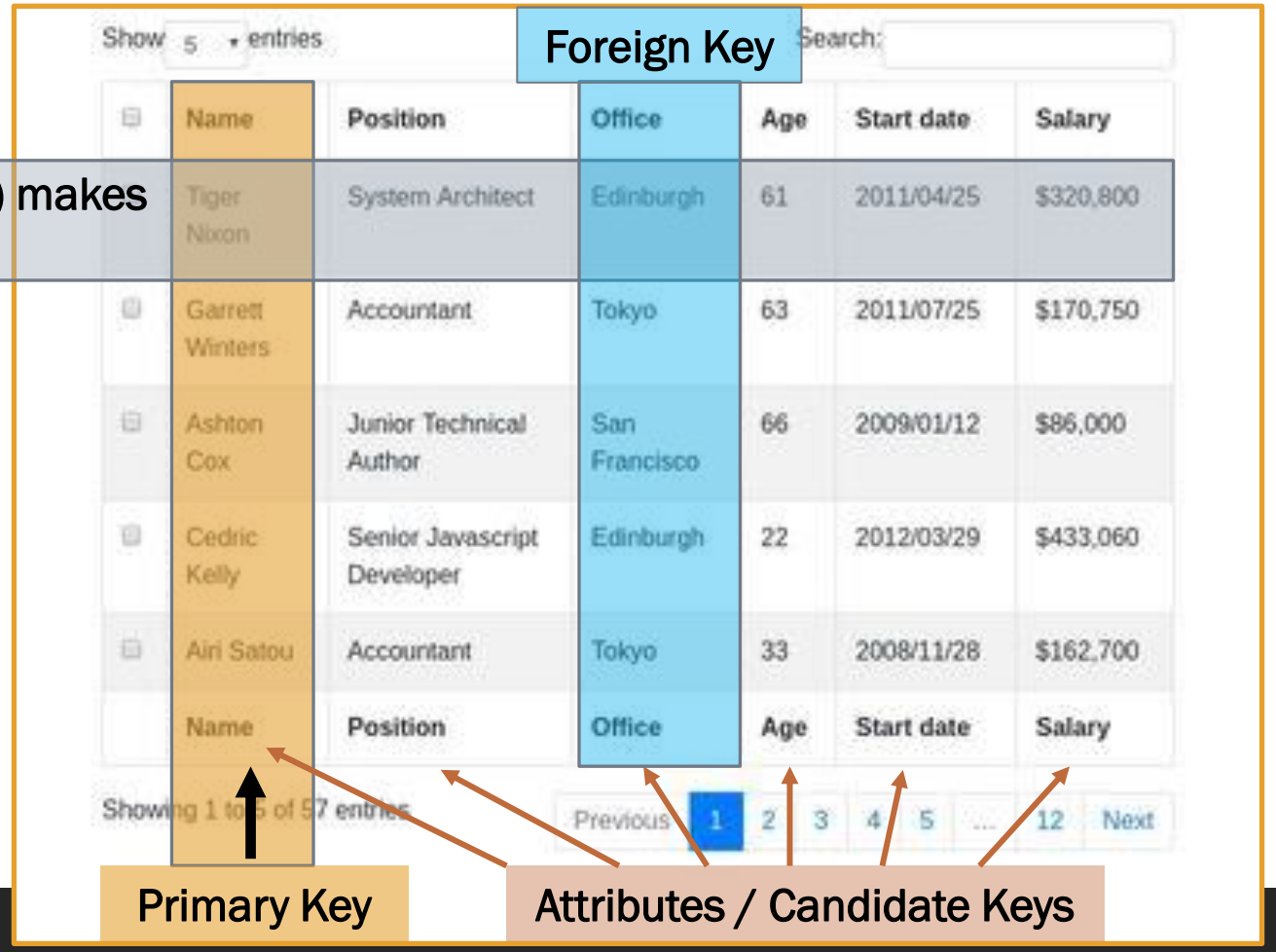
Databases – Instances/Users

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***. →→→→→



Database Management System (DBMS)

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 data that is related by its characteristics.
- A Database Management System (*DBMS*) stores data so it's easy to insert and retrieve.

If you have the grades of all students, you can easily extrapolate average grades, highest grades, GPA, etc.

Primary Key
*Not a good example!

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

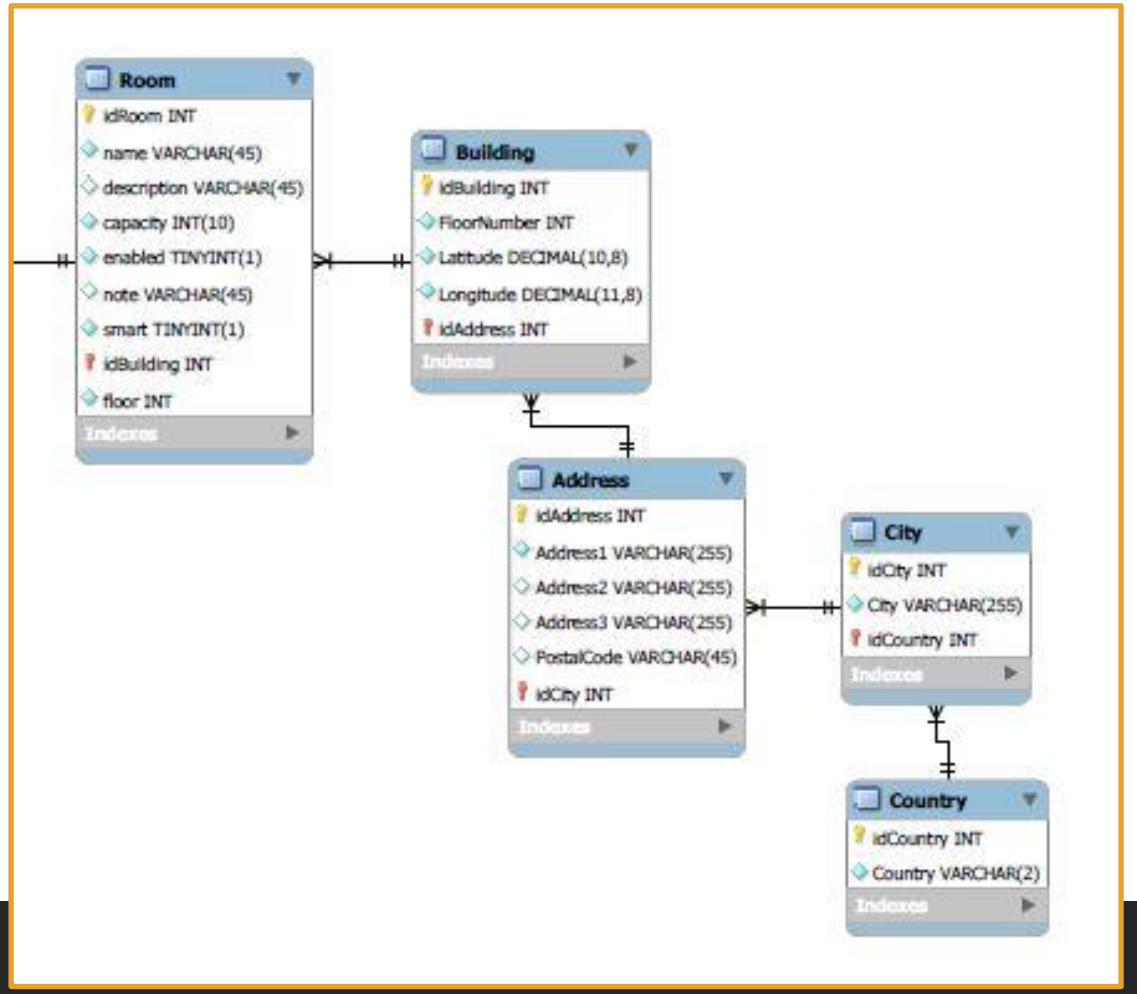
Database Schema Diagram

https://www.tutorialspoint.com/dbms/dbms_data_schemas.htm
https://en.wikipedia.org/wiki/Data_modeling

A database ***schema*** is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and the ***relations*** among them.

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

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

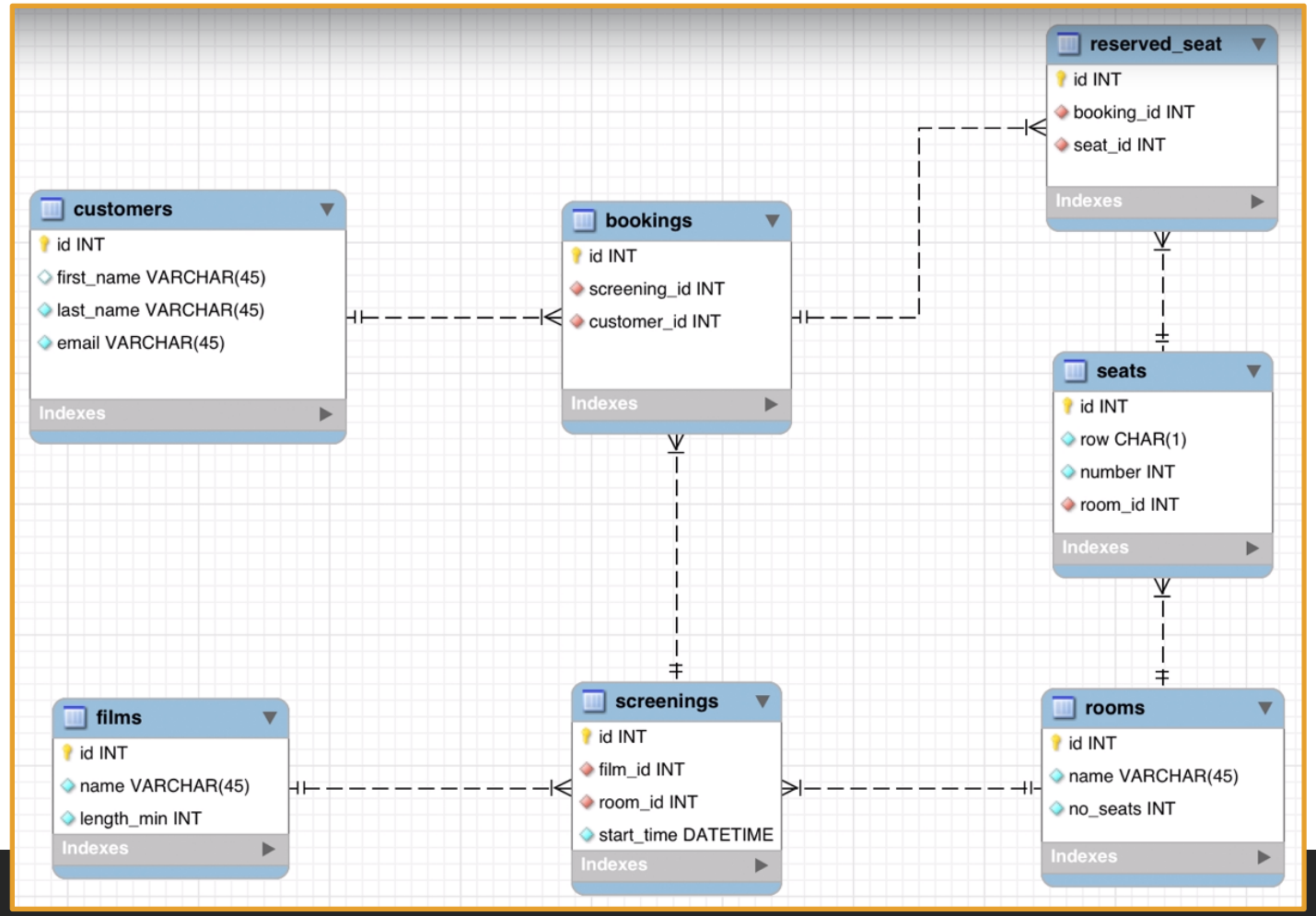


Database Schema Diagram

https://www.tutorialspoint.com/dbms/dbms_data_schemas.htm
https://en.wikipedia.org/wiki/Data_modeling

Red Color means FK

- Solid Shape == NOT NULL Constraint
- Empty Color Outline == NULL
- Golden key == PK
- Red Key == FK (also a PK in another table).
- Blue Diamond == NOT NULL Constraint
- Red Diamond == NOT NULL and is a FK.

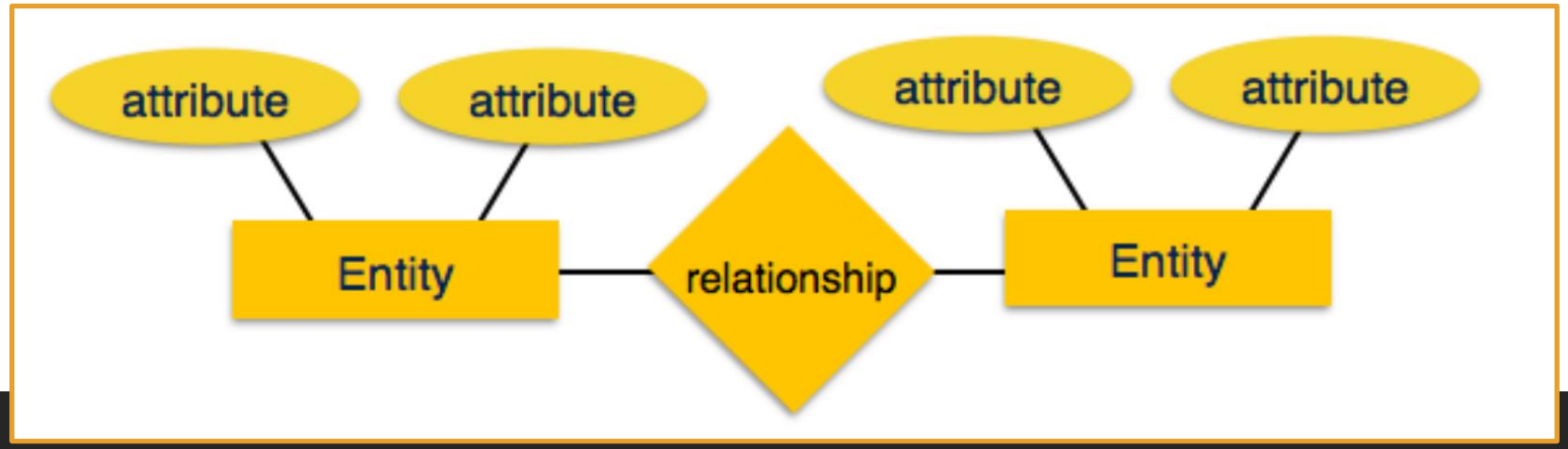


Entity-Relationship Models

https://www.tutorialspoint.com/dbms/dbms_data_models.htm
https://en.wikipedia.org/wiki/Data_modeling

Data models define the structure of a database. ***Data Models*** are ***entities*** that abstract the data from a ***DBMS*** into a more human readable format. ***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.

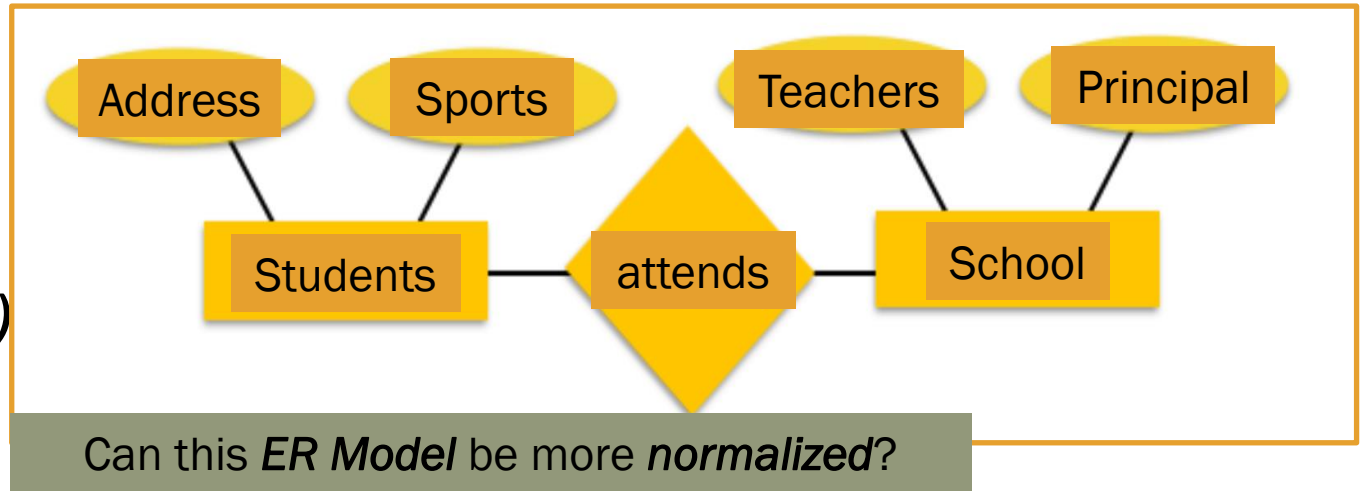


Entity-Relationship Models

https://www.tutorialspoint.com/dbms/dbms_data_models.htm
https://en.wikipedia.org/wiki/Data_modeling

- **Entity** – a real-world thing having properties called **attributes**.
- **Attribute** - the details about an ***entity***.
- **Relationship** – The logical association between entities. Relationships are mapped with entities in 3 ways.
- "Mapping cardinalities" defines the number of associations between two entities.

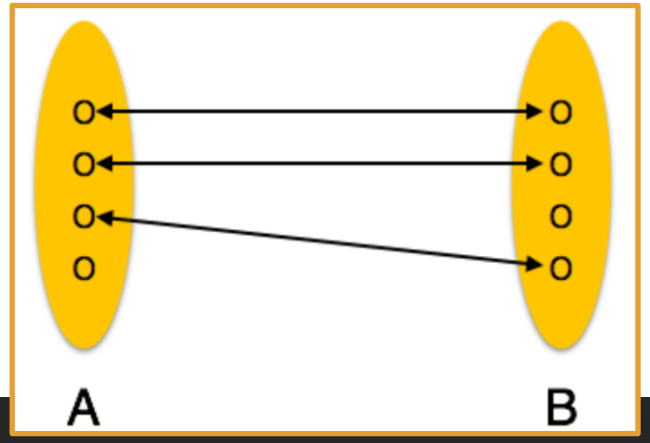
- Mapping cardinalities
  - *one to one*
  - *one to many (1-N, N-1)*
  - *many to many*

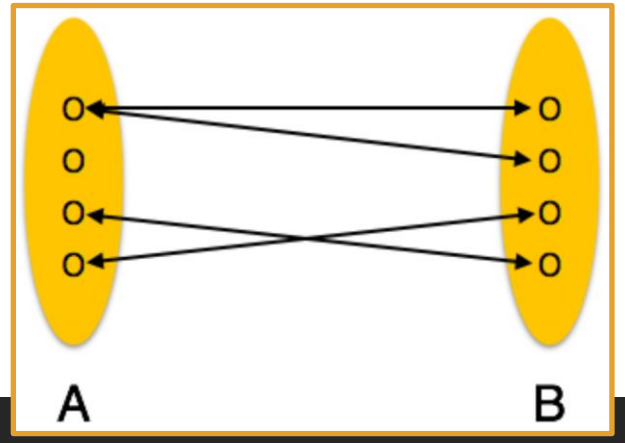


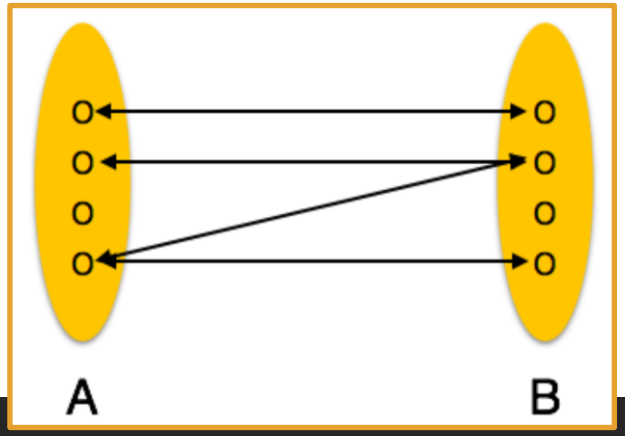
Relational Database – Relationships

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

Multiplicity		
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. Use a Cross-Reference Table.







Cardinality vs Multiplicity

https://martinfowler.com/bliki/MultiplicityNotCardinality.html
https://www.tutorialspoint.com/dbms/er_model_basic_concepts.htm

---

When data modeling methods talk about relationships, they use the term ***cardinality*** to indicate how many entities may be linked together. You might have a relationship between order and customer and say that the cardinality of the relationship is one-to-many. Or you might hear that the cardinality of customers for an order is 0-to-many.

UML avoids the term cardinality preferring to use *multiplicity*. Often people with a data modeling background are surprised at this since cardinality has been so widely used in data modeling circles.

The reason for the change is that the dictionary definition of cardinality is "*the number of elements in a particular set or other grouping*" (OED). According to this the data modeling usage is actually wrong. In the excellent UML Reference Manual, Rumbaugh defines multiplicity as "A specification of the range of allowable cardinality values - the size - that a set may assume". The UML uses multiplicity in various places, for a property (association or attribute) and also to show the multiplicity of parts in a composite structure. It's formally defined as a lower and upper bound. An association (the UML equivalent to a relationship in data modeling circles) has a multiplicity for each direction.

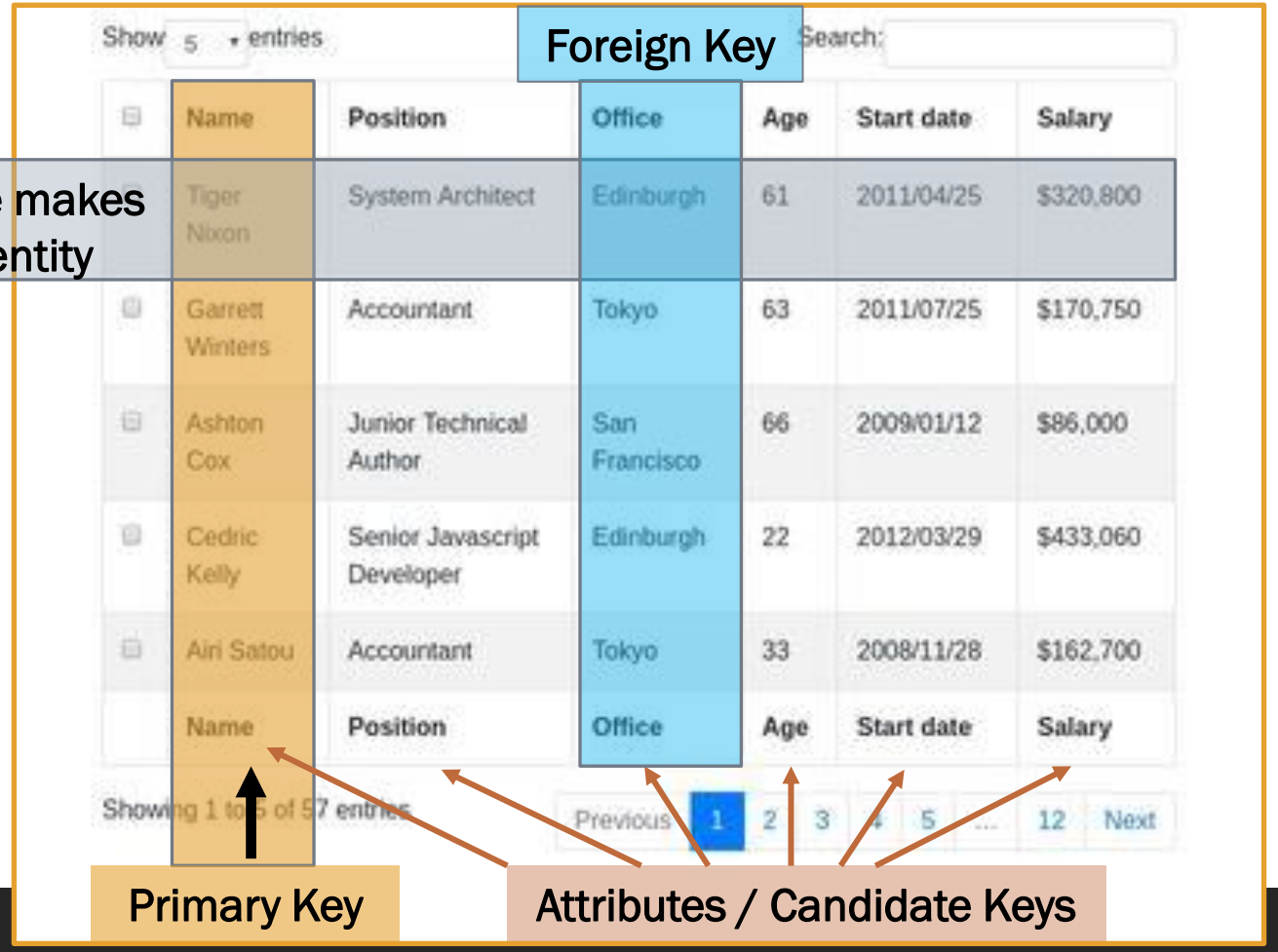
Primary and Candidate Keys

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***).

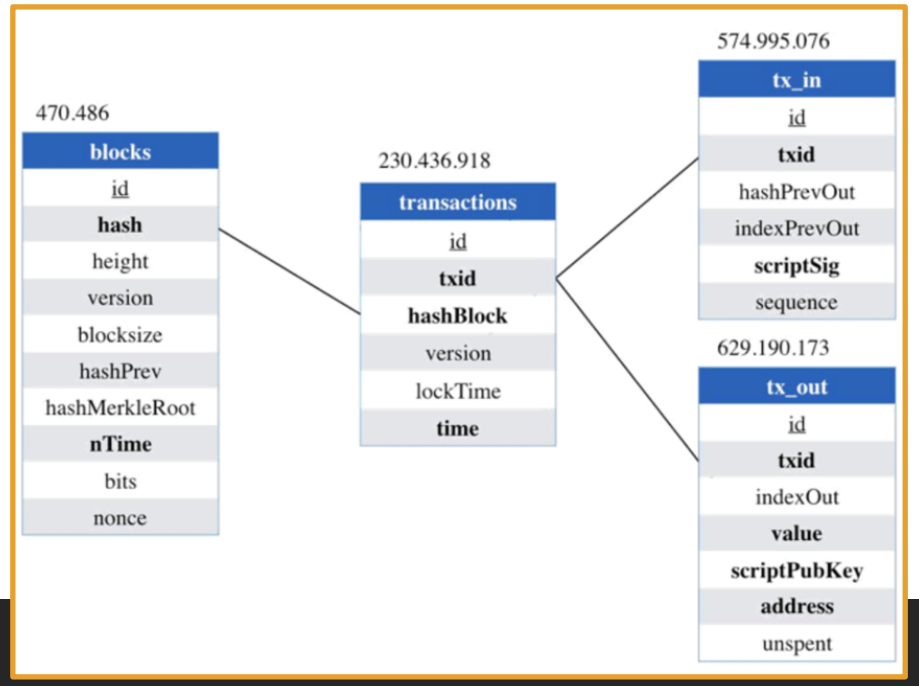
Foreign Key – The ***Primary Key*** of another table. Used to represent to other entity.



Relational Database

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

A ***RDBMS*** allows ***entities*** to form ***tables*** with ***relations*** between them. You could ideally 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 Database 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

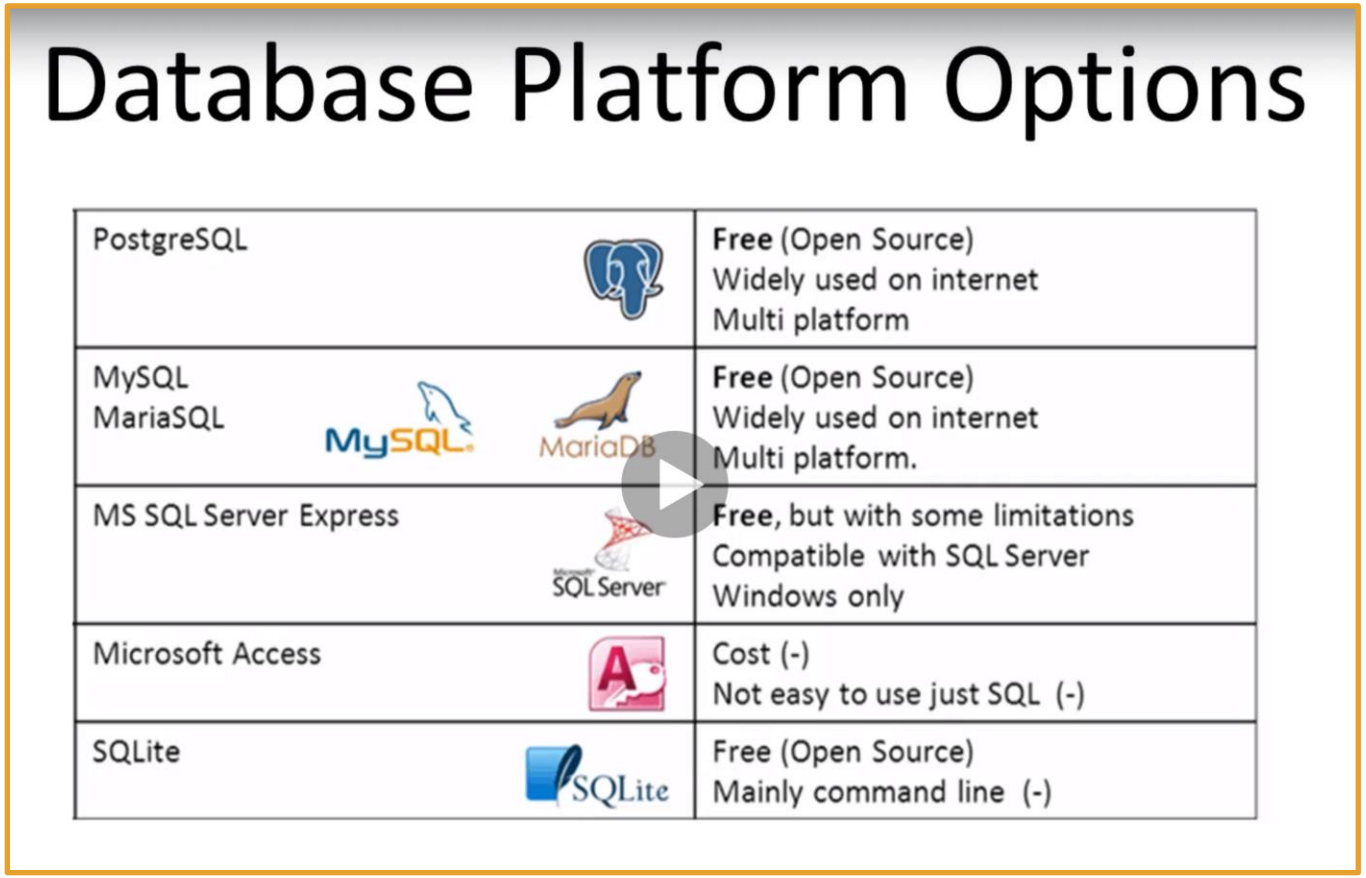
https://www.tutorialspoint.com/dbms/database_normalization.htm

---

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.

DataBase Platform Options

There are many options when it comes to choosing your next Database.