DSD Theory Cheatsheet

Thomas Boxall
up2108121@myport.ac.uk

May 2023

1 Data

Data is facts and statistics collected together for reference or analysis.
Information is the result of processing data by adding context and meaning. We should carefully examine the context before drawing information from data.

2 Database Management System

The *DataBase Management System* (DBMS) is the core of the database system. All communication to the database goes through DBMS.
DBMS controls users, access to data & schema and provides an view of operations. DBMS removes risk of inconsistent data & improves the ease which security can be controlled with.
Two different languages are used to interface with a database. *Data Definition Language* (DDL) defines the structure of the database. *Data Manipulation Language* (DML) is used to manipulate the data.

3 Entities & where we keep them

Entity a 'thing' about which data is stored (often will be nouns). These should be singular, not plural.
Attribute a quality of an entity (eg id number, name, etc)
Table A 2D representation of entities. (AKA relation table or relation).
Record a row of data in a table. (AKA tuple or row).
Relationship is how two tables are related to each other, they are represented in relations.
When designing a database we have to think about the entities we will store, what data about them we need, and what form this data will take.

3.1 Entity Relationship Diagrams

ERD a diagram showing how entities & attributes are related. They are built from business rules (*statements that define how companies do things*). Entities are joined up using 'Crows Foot Notation' (Figure 1).

4 Keys

Primary Key an attribute (or combination of attributes) which uniquely identifies a record. Every table must have one.
Composite Key a combination of attributes (each called simple keys) that act as a primary key in a table.
Foreign Key an attribute (or combination of attributes) which is a primary key in another table.
Alternate Key gives an alternate access path to data that is not via the primary key. This is *bad* and if correctly normalised, this shouldn't exist.

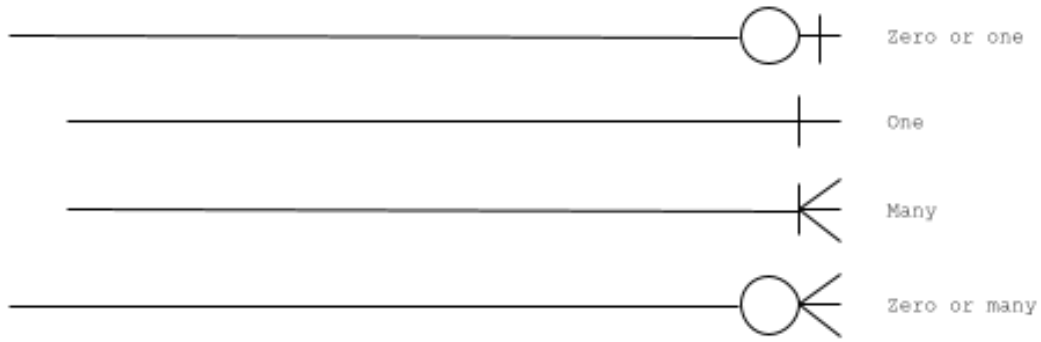

Figure 1: Crow's foot notation

5 Attributes

Constraint a rule that protects your data or enforces certain behaviour.
All the different bits of data we store about an entity will all be different attributes. The data we store has to be GDPR compliant (must be: adequate, relevant and limited to what is necessary).

5.1 Data Types

see my notes for more details on each data type.
Numerical smallint, integer, bigint, decimal, real, double, serial, bigserial
Alphanumeric text, char, varchar
Date & time timestamp without timezone, timestamp with timezone, date, time without timezone, time with timezone.

6 Normalisation

Normalisation the process of designing a database in a way that reduces data redundancy and makes the database more efficient.
Data Redundancy where there is repeated data, for example storing the full details of each boatyard with each employee.
There are many *normal forms*, we are only concerned with 1st through 3rd.

6.1 First Normal Form

- Each column should contain single (atomic) values (each column should not hold more than one value)
- Values stored in a column should be of the same domain (data type)
- All the columns in a table should have unique names
- The order in which the data is stored doesn't matter

Quite often when normalising to 1NF, you will find that a new entity gets created.

6.2 Second Normal Form

- Be in 1NF
- Have no partial dependencies (where a part of a record can be identified by something other than the primary key)

6.3 Third Normal Form

- Be in 2NF
- Have no transitive dependencies (where an attribute is dependent on an attribute which is not the primary key)

7 Joins

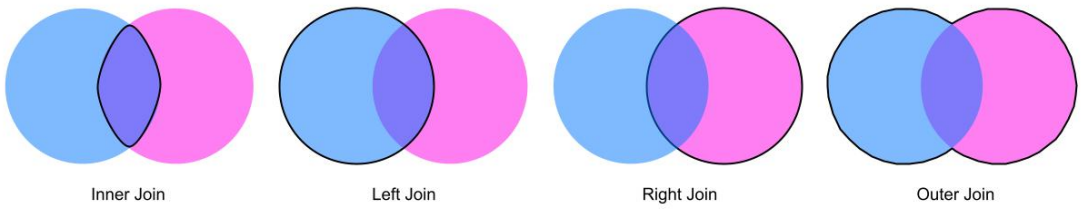

Figure 2: Joins

Joins allow us to join multiple tables together to get more data out of the database from a single query. Figure 2 shows the different types of joins.
Cartesian Product is the result of a bad join where the DBMS will return every record in one table is joined to every record in a second table. To avoid this, we tell the DBMS what attributes to match.
Inner join returns all the data where there is overlap in both tables. The overlap is decided base on the joining attribute.
Left join returns all the records in the left table with the entries of the right table where there is a match on the joining attribute.
Right join returns all the records in the right table with the entries of the left tables where there is a match on the joining attribute.
Full outer join returns everything from the two tables.
When joining two tables together, remember to use the correct type of join and to match the joining attribute (remember these may have different names in different tables).

8 Order Of Execution

When executing a query, the DBMS will run statements in a different order to that which we enter them in.

1. `FROM` & `JOIN` (choose and join the tables to get base data)
2. `WHERE`, `SUBQUERY`, `INTERSECTION`, `UNION`, `EXCEPT` (filters the base data)
3. `GROUP BY` (aggregates the base data)
4. `HAVING` (filters the aggregated data)
5. `SELECT` (returns the final data, not displayed yet)
6. `ORDER BY` (sorts the final data)
7. `LIMIT` (limits the returned data to a row count)
8. display data

9 Security

To steal data, you just have to make a copy of it. The biggest security risk is those who have access to it.
Superuser is the top-level administrator in PostgreSQL. These accounts can do *everything* so giving someone this role should be minimised as much as possible.

9.1 Roles

Before a user (role) can log into the DBMS, they have to have a role created and own a database with the same name as their username. Passwords must be set at the time of user creation, users should change passwords frequently. Once a user has been created, they have no permissions other than login. You have to grant permissions to users to perform actions.
Views can be used as a way of limiting user access to data. They allow pre-written queries to be executed which can only show *some* data.

9.2 Encryption

By default (out of the box), encryption in PostgreSQL is disabled. It must be enabled before use.
A number of different hashing and data security options are available, including: PGP, md5, sha1, and sha255.
Salt can be added to data which is being encrypted to ensure every encrypted value is different, even if the values to be encrypted are the same.