

Database:

A set of related data and the way it is organized

By a Database Management System: computer software that allows users to interact with database and provide access to all of the data.

Importance:

12 rules of what makes a database management system a true relational system.

- Rule 1: information rule

All information in a relational database is represented explicitly at the logical level and in exactly one way - values in tables.

→ Standardization of data model

→ flexibility in adding and altering tables.

→ Data integrity

→ Standard Query Language (SQL)

→ Simplicity

→ Intuitive Organization

OLAP vs OLTP:

↳ OLAP: Online Analytical Processing

→ Databases optimized for their workload allow for complex analytical and ad hoc queries. These type of databases are optimized for reads.

as a OLTP Online Transactional Processing → Database optimized for these workloads allow for less complex queries in large volumes. The type of queries for this database are read, insert, update and delete.

Structuring the Database:

Normalization → To reduce data redundancy and increase data integrity

The process of structuring a relational database in accordance with a series of normal forms in order to reduce data redundancy and increase data integrity.

Denormalization → Must be done in read heavy workloads to increase performance.

Objectives of Normal Form:

1. To free the database from unwanted insertions, updates and deletion dependencies
2. To reduce the need of deflating the database as new types of data are introduced
3. To make the relational model more informative to users.
4. To make the database neutral to the query statistics

Normal Forms:

First Normal Form (1NF)

- Atomic values: each cell contains unique and single values → No sets, collections, lists
- Be able to add data without altering tables → By creating new tables (and relate them)
- Separate different relations, into different tables → Customer table, Sales table (not anyone)
- Keep relationships between tables together with foreign keys → By using attributes present in both tables

Second Normal Form (2NF)

- **First 1NF**
 - All columns in the table must rely on the = Primary Key '
↳ Primary key needs to be formed by a single "column"

Table 1	Table 2
Store ID	Cost ID

~~Other Info~~

Q: We need to know 2
"keys" before being able
to have unique can of
= Old info".

↳ Solution: Split into 2 tables

Third Normal Form (3NF)

- Have ZNF
 - No "transitive dependency"

C, To get from A to C, you want to avoid going through B

Transition dependency

6

Transitive step

Maria's Hand	Year	Band	Lead Singer
Pizzello's Talk			

Bubbles to apply by concept

Band Singer's Table

Allots us to update data just in one place.

ff Renormalization:

SQLs on the database allow for outstanding flexibility but are extremely slow.

If you are dealing with heavy reads on your database, you may want to think about denormalizing your tables.

\Rightarrow Normalize \Rightarrow Denormalize (comes after normalization)

The process of trying to improve the read performance of a database at the expense of losing write performance by adding redundant copies of data.

→ Requires more space in system

→ Changes in processing speed → Reads (Fast) / Writes: insert, update, delete (Slow)

Fact and Dimensions Tables

→ Work together to create an organized data model

Fact Tables: consists of the measurements, metrics or fact of a business process
→ customer transactions, ...

Dimension Tables: a structure that categorizes facts and measures in order to enable users to answer business questions. Dimensions are people, product and time.

A A dimensional data element is similar to a categorical variable in statistics

↳ Provides structure labeling information to otherwise unstructured numeric measures.

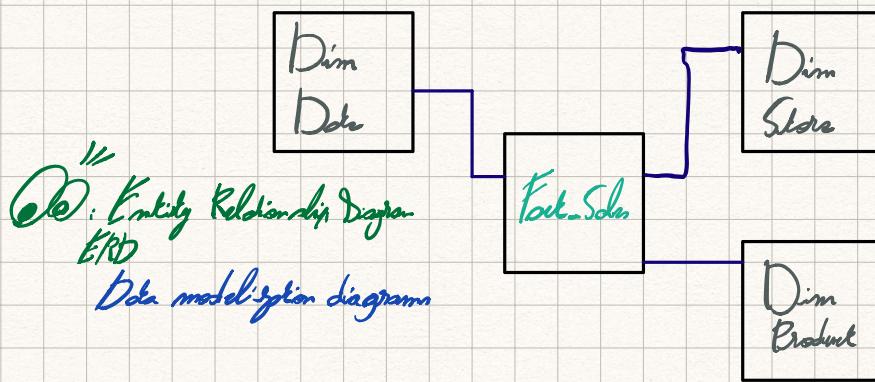
66

The fact table contains business facts (or measures) and foreign keys which refer to candidate keys (normally primary key) in the dimension tables.

99

Contrary to fact tables, dimension tables contain descriptive attributes (fields) that are typically textual fields.

A Fact table is located at the center of the star or snowflake schema surrounded by dimension tables.



Dimension tables:

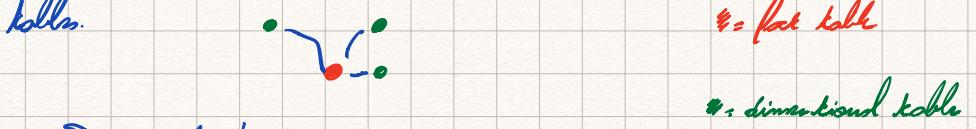
- Where (Dim_Store)
- When (Dim_Date)
- What (Dim_Product)

Fact Table.

How many? Business Brain Model

Implementing Different Schemas:

Star Schema: one or more fat tables referencing any number of dimensional tables.



Benefits:

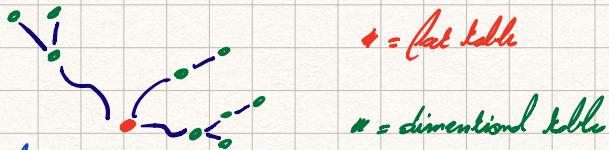
- Denormalized
- Simplifies queries (relaxation of 3NF)
- Fast aggregations

Issues:

- Issues that come with denormalization
 - Data integrity
 - Decrease query flexibility
- Many-to-many relationships -- simplified

Snowflake Schema:

→ More complex schema.



• = fact table

• = dimensional table

→ Allows one-to-many relationships

→ Snowflake schema more normalized than "Star schema" but only in 1NF or 2NF

Data definition and constraints:

Creating statements in SQL has a few important constraints highlighted below:

• - NOT NULL: column can not contain null values.

```
CREATE TABLE IF NOT EXISTS ... (
    customer_id int NOT NULL,
    store_id int,
    ...
);
```

"UNIQUE": data across all rows must not be repeated; can be combination of columns.

```
CREATE TABLE IF NOT EXISTS ... (
    customer_id int NOT NULL,
    store_id int,
    UNIQUE (customer_id, store_id, open))
```

PRIMARY KEY: is unique and non-null field used to identify the rows in the table; can be composition of several columns (composite key).

```
CREATE TABLE IF NOT EXISTS ... (
    customer_id int NOT NULL,
    store_id int,
    PRIMARY KEY (customer_id, store_id))
```

Update: updating or inserting a new row

INSERT: add new row `INSERT into table_1 (`

VALUES

```
(432, -758 Main Street, 'Chicago', 'IL');
```

→ ON CONFLICT DO NOTHING

```
INSERT into table_1 (
```

```
VALUES (432, -758 Main Street, 'Chicago', 'IL')
```

```
ON CONFLICT (customer_id)
```

```
DO NOTHING;
```

Update if exists pattern definition

→ ON CONFLICT DO UPDATE

```
INSERT into table_1 (
```

```
VALUES (432, -758 Main Street, 'Chicago', 'IL')
```

```
ON CONFLICT (customer_id)
```

```
DO UPDATE
```

```
SET customer_street = EXCLUDED.customer_street
```