

Database Normalization

Informal Design Guidelines

1) Semantic of attributes in relations:

Design a relation schema so that it is easy to explain its meaning.

Do not combine attributes from multiple entity types and relationship types into a single relation.

2) Redundant information and update anomalies

Design the base relation schemas so that no insertion, deletion, or modification anomalies are present in the relations.

If any anomalies are present, note them clearly and make sure that the programs that update the database will operate correctly

3) Null Values in Tuples

As far as possible, avoid placing attributes in a base relation whose values may frequently be NULL. If NULLs are unavoidable, make sure that they apply in exceptional cases only and do not apply to a majority of tuples in the relation.

4) Generating Spurious Tuples

Design relation schemas so that they can be joined with equality conditions on attributes that are appropriately related (primary key, foreign key) pairs in a way that guarantees that no spurious tuples are generated.

Avoid relations that contain matching attributes that are not (foreign key, primary key) combinations because

joining on such attributes may produce spurious tuples.

Types of Normalization in SQL

Types of Normalization in SQL



1NF

First Normal Form

- Ensures that each column contains only atomic values



2NF

Second Normal Form

- Eliminates partial dependencies.



3NF

Third Normal Form

- Eliminates transitive dependencies.



BCNF

Boyce-Codd Normal Form

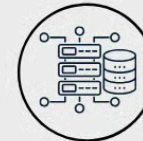
- Strict version of 3NF that addresses additional anomalies.



4NF

Fourth Normal Form

- Deals with multi-valued dependencies



5NF

Fifth Normal Form

- Addresses join dependencies

First Normal Form (1NF)

- **Atomic values only**
 - Each column must contain **indivisible (single) values**.
 - No lists, arrays, or multiple values in one cell.
- **One piece of information per cell**
 - Every cell holds **exactly one fact**, like a spreadsheet cell.
- **Unique column names**
 - Each column must have a **distinct name**.
- **Goal**
 - Ensure **data atomicity** and a clear, consistent table structure.

يعني من الاخر كل عمود لازم تكون القيم الى فيه هي قيمه واحده مش قيمه متكونه من اكثر من جزء تقدر تجزئها على اكثر من صف .

Second Normal Form (2NF)

- **Removes partial dependencies**
 - Non-key attributes must not depend on part of a composite key.
- **Direct dependency on the primary key**
 - Every non-key column depends **only** on the full primary key.
- **No column-to-column dependency**
 - Attributes should not depend on other non-key attributes.

"لازم كل العواميد تعتمد على المفتاح كله مش جزء منه بس"، ولو فيه عمود معتمد على 'حته' بس، نفصله في جدول لوحده."

Third Normal Form (3NF)

- **Eliminates transitive dependencies**
 - Non-key attributes must not depend on other non-key attributes.
- **Direct dependency on the primary key**
 - Every non-key attribute depends **only** on the primary key.
- **Builds on Second Normal Form (2NF)**
 - A table must already satisfy 2NF requirements before reaching 3NF.

"لازم كل العواميد اللي مش مفاتيح تعتمد على المفتاح الأساسي مباشرة، وماينفعش عمود (مش مفتاح) يحدد معلومة لعمود ثاني (مش مفتاح) زيه؛ يعني ممنوع 'الواسطة' بين العواميد العادية. بس الـ 3NF بتسمح بحالة واحدة: إن عمود 'مش مفتاح' يحدد معلومة لعمود ثاني، بشرط إن العمود الثاني ده يكون أصلاً 'جزء من مفتاح مركب' موجود في الجدول."

Boyce-Codd Normal Form (BCNF)

- **Stricter than 3NF**: Builds on Third Normal Form by removing remaining anomalies.
- **Key rule**: Every determinant must be a **candidate key**.
- **Purpose**: Eliminates redundancy caused by functional dependencies that 3NF may still allow.
- **Result**: More robust schema with fewer update, insertion, and deletion anomalies.

"هنا مفيش استثناءات؛ القاعدة بتقول إن أي عمود 'بيتحكم' في معلومة عمود غيره $(X \rightarrow Y)$ ، لازم الـ (X) ده يكون هو 'مفتاح مركب' (Candidate Key) للجدول. لو فيه عمود بيحدد معلومة غيره وهو مش مفتاح، يبقى لازم تفصلهم في جدول جديد وتخلي العمود اللي كان بيحدد ده هو 'المفتاح الأساسي' هناك."

3NF VS BCNF

المرحلة	القاعدة	بتسمح بإيه؟
3NF	تمنع الاعتماد بالواسطة) $(NonKey \rightarrow NonKey)$.	بتسمح إن $(NonKey)$ يحدد عمود $(Prime)$ (جزء من مفتاح).
BCNF	تمنع أي $(NonKey)$ يحدد أي حاجة ثانية خالص.	مابتسمحش؛ لازم اللي يحدد يكون $(CandidateKey)$ ويس.