

## **GrantGuru Normal Forms Discussion**

Abdullahi Abdullahi, Mathieu Poulin, James Tedder, Colby Wirth

October 15, 2025

## Highest Normal Form Achieved

The GrantGuru database design satisfies **Boyce-Codd Normal Form (BCNF)** but does not achieve Fourth Normal Form (4NF). Our design eliminates partial dependencies, transitive dependencies, and ensures all determinants are superkeys. However, multi-valued dependencies in the research fields' attributes prevent 4NF compliance.

## How the Database Achieves BCNF

### First Normal Form (1NF): Atomicity

The database achieves 1NF through atomic attributes and unique identification. The **User entity** properly decomposes the composite **Name** attribute into **f\_name**, **l\_name**, and **m\_name**. All attributes (**user\_id**, **email**, **hashed\_password**, **institution**) contain single, indivisible values. The **Grant entity** maintains atomicity with single-valued attributes like **grant\_id**, **grant\_title**, **award\_max\_amnt**, and **provider**. The **Documents entity** stores atomic values for **document\_id**, **document\_name**, and **document\_type**. Each entity has a primary key ensuring unique record identification.

### Second Normal Form (2NF): No Partial Dependencies

The design satisfies 2NF by using single-attribute primary keys, preventing partial dependencies. In the **Application entity**, **application\_id** serves as the primary key, and the non-key attribute **status** depends on the complete key. Since there are no composite primary keys, partial dependencies cannot exist. In the **Documents entity**, attributes like **document\_type** depend entirely on **document\_id**, not on any portion of a composite key.

### Third Normal Form (3NF): No Transitive Dependencies

The database achieves 3NF by ensuring direct dependencies without transitive relationships. In the **Grant entity**, all attributes depend directly on **grant\_id**: **grant\_id** → **provider**, **grant\_id** → **eligibility**, **grant\_id** → **award\_max\_amnt**. Importantly, **award\_max\_amnt** and **award\_min\_amnt** are independent attributes both determined by **grant\_id**, not by each other, avoiding transitive dependency. In the **User entity**, **institution** depends directly on **user\_id**, not transitively through **email**. The **Application entity** demonstrates 3NF as **status** depends solely on **application\_id** without intermediate non-key attributes.

### Achieving BCNF: All Determinants Are Superkeys

BCNF requires that every determinant must be a superkey. The GrantGuru database satisfies this:

**User Entity:** Primary Key: **user\_id**; Candidate Key: **email**. Both determinants are superkeys.

**Grant Entity:** Primary Key: **grant\_id**. The only determinant is the primary key.

**Application Entity:** Primary Key: **application\_id**; Potential Candidate Key: (**user\_id**, **grant\_id**). Both are candidate keys.

**Documents and Internal Deadlines:** Single primary keys (**document\_id**, **deadline\_id**) serve as sole determinants.

There are no functional dependencies where a non-superkey determines other attributes—every functional dependency has a superkey on the left side.

## Why the Database Does Not Achieve 4NF

Fourth Normal Form eliminates multi-valued dependencies (MVDs). An MVD  $\alpha\beta$  means that values of  $\beta$  associated with a given  $\alpha$  are independent of other attributes. However, the GrantGuru database contains a MVD violation:

### Multi-Valued Dependency in User Entity

The `research_fields` attribute creates: `user_id research_fields`

A faculty member has multiple independent research interests (e.g., 'Machine Learning,' 'Computer Vision,' 'NLP'). These are independent of `email` and `institution`. If stored as multiple rows, `email` and `institution` are redundantly repeated for each research field, violating 4NF.

### Justification for Not Achieving 4NF

**Implementation Simplicity:** Avoiding separate research field tables accelerates initial deployment while preserving future normalization options.

### Path to 4NF

To achieve 4NF, decomposition would be required:

```
User(user_id, email, hashed_password, f_name, l_name, m_name, institution)
UserResearchField(user_id, research_field)
```

This would eliminate redundancy and satisfy 4NF by removing the multi-valued dependencies.