

10/15/18 Notes

Design Principles

Referential Integrity - a property of a database enforcing foreign keys to reference only existent primary keys

- **Insert Anomaly** - the dependence of a piece of information existing for some other information to be added

- *Ex.* For a database with only this table,

employeeld	name	office	office_location
------------	------	--------	-----------------

we cannot add a new office unless we add a new employee.

- **Delete Anomaly** - the forced removal of a piece of information when removing some other information

- *Ex.* For the table and database described above,

employeeld	name	office	office_location
------------	------	--------	-----------------

removing an employee may remove information on an entire office

- **Update Anomaly** - the inconsistency resulting from updating tables that contain the same redundant information in another table

- *Ex.* For the tables below,

employeeld	name	office	office_location
------------	------	--------	-----------------

office	office_location	phone_number
--------	-----------------	--------------

Updating the office location of an office in the second table may result in a conflict with the first table if you don't also update the location of the office there as well.

- **Normalization** - A way of structuring a database to reduce the most amount of anomalies and uphold the integrity of information.

- Normalization is usually done up to a *form*:
- **First Normal Form (1NF)** - all fields in the database have scalar values.

- *Ex.* This table violates 1NF:

id	name	pokemon
1	Ash Ketchum	Pikachu, Charizard, Bulbasaur
2	Barack Obama	Geodude, Onyx

The reason this table violates 1NF is because the field "pokemon" has values that contain more than one Pokemon.

- **Second Normal Form (2NF)** - all tables in the database obey 1NF, and also no fields are functionally dependent on a proper subset of the primary key.

- **Functional Dependency** - the dependence of one attribute on another

- And for people that don't care about math, **proper subset** - a subset of another set that is not equal to that set.
- *Ex.* This table violates 2NF:

restaurant	food	ingredient	price
Chick-Fil-A	chicken sandwich	chicken	\$3.05
McDonalds	mcflurry	ice cream	\$1.79

The reason this table violates 2NF is because the primary key is clearly {restaurant, food}, however the ingredient only depends on part of the primary key - the food part.

- **Third Normal Form (3NF)** - all tables in the database obey 2NF, and also all fields in a table are dependent *only* on the primary key and no other keys.
 - *Ex.* This table violates 3NF:

id	name	house_number	house_color
1	Jason	1023	white
2	Robert	1027	red

The reason this table violates 3NF is because the field "house_color" depends on the field "house_number", which is not the primary key.