***Udemy – SQL***

***Section 6 – Fundamentals of Database Theory***

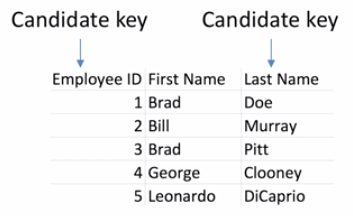
*Structure vs. Unstructured Data*

* **Structured** 🡪 simple table w/ attributes + tuples/values w/ proper relations that don’t contradict any user-set rules or database rules
* *Proper SQL tables*
* Power of structured data = simplicity to perform any set of analysis + results
* Advantages:
* Data consistency
* More info from the same data
* Improved data integrity (w/ correct rules + no input errors)
* Improved security 🡪 DB’s are smaller due to text-based files (compared to music or movies) but we can assign a lot of security checks w/out sacrificing DB efficiency
* Improved maintenance 🡪 less disk space taken up by DB’s = quicker query results + DB checkups
* Offers more in terms of storing and analyzing data so organizations will keep it around, even w/ growing popularity of unstructured data (will never sacrifice data efficiency and integrity)
* **Unstructured** 🡪 music, movies, pictures, streaming text (very RICH data)
* *Everything else (cannot be represented in a database)*
* Makes up 80% of data out there
* Almost impossible to manipulate + analyze + get results out of it w/out specialized tools

*Elements of a DB*

* Consists of **tables** + **schemas** (manage dependencies between different tables), as well as **catalogs** and **extensions** to store different functions
* **Tables**/Relations consist of **fields/attributes**/columns and **tuples/records**/rows
* **Degree** = # of attributes w/in a table
* **Cardinality** = # of rows w/in a table
* Can *also* refer to unique data w/in any columns, or w/in the whole relation/table
* i.e. unique constraint on a column/cell = cardinality for the table
* **Domain** = set of allowable values for 1+ columns (ex: 0 to infinity for copies of games sold)
* Every table has **primary** and **foreign keys**
* **PK =** unique identifier of a row w/in a table
* **FK** = connect/relates 2 tables

*Relational Keys*

* **Superkey** = any combo of columns that uniquely ID’s a row
* **Candidate key** = a superkey such that no subset of it can be a superkey
* Has **uniqueness + irreducibility**
* 
* **PK =** candidate key selected to uniquely ID tuples/rows/records w/in a relation/table
* If table/relation has several candidate keys, only 1 can be the PK
* **FK =** attribute/set of attributes w/in 1 relation/table that matches the candidate key/PK of some other relation

*Relational DB*

* **RDB** = digital DB whose organization is based on the relational model of data/ a set of tables
* Popular due to usability + accessibility done via enforcing different rules + constraints + due to their consistency + availability
* Some limitations 🡪 any change requires extensive work to satisfy all existing conditions (+ possible create new ones)
* There are also some DB usage + storage limits
* that satisfies the following data integrity
* **Entity integrity** = no duplicate rows in a table (can share *some* attribute values, but not be *exactly* the same) 🡪 PK’s and CK’s enable this
* **Domain integrity** 🡪 enforces valid cell entries for a given column by restricting type, size, performance, and/or range of possible values
* **Referential integrity 🡪** rows being used by other records in other tables cannot be deleted
* Can override and delete but this causes a damaged + possibly corrupted DB
* **User-defined integrity 🡪** enforces specific business rules that don’t fall into domain or referential integrity (i.e. some attribute value for our company must be NOT NULL, even if this isn’t required for the DB to work)

*Functional Dependency*

* Main reason for using SQL and RDB’s is to assign dependencies between tables and between data
* If we can relate tables, then the amount of analysis we can do grows incredibly compared to analyzing tables on their own
* **Dependencies** occur when a attribute/set of attributes ID’s a particular value of another attribute
* When designing efficient DB’s that avoid redundancy, IDing dependencies can help us ensure every column in a table is dependent on the PK + that the PK is as simple as possible