CSI 360 - Database Systems - Exam Review

DB Ideas

1. Lots of data
2. meta-data / self description
3. easy to use / share / view
4. data redundancy and control
5. algorithmic help - sorts/constraints
6. efficient retrieval of information
7. multiple users
8. security and protection
9. concurrency control
10. transaction control
11. backup and recovery
12. storage and distribution

What are the main roles?

1. end users (they don't need to know about DBs at all)
2. designers (when you're figuring out the tables and how to put stuff in / pull it out)
3. administrators (what's the architecture? what hardware? How is it actually stored?)
4. implementors (Who wrote this software in the first place?)

What's ACID?

1. A - atomicity - transaction processing - everything happens all at once
2. C - consistency - everything remains ”valid”
3. I - isolation - Things work and are isolated from one another (the 10 to each other example)
4. D - non-volatile storage after commits

Entity Relationship (ER) diagrams

1. Squares are entities
2. Circles are attributes
3. PKs are underlined (PK below)
4. Diamonds relate two entities
5. `||` means one and only one mandatory
6. `|0` means 0 or 1
7. `>|` means one or more
8. `>0` means 0 or more
9. Usually, every entity becomes a table in a DB
10. Usually, Many to Many Relationships become Junction Tables in a DB
11. Usually, other relationships become columns added to a table
12. Access has a number of types. Refer to:
    `https://www.w3schools.com/sql/sql_datatypes.asp`

Brewer's Theorem (Cap Theorem)

A DB can never guarantee (at all times) more than two of the following

1. Consistency - every request gets the most recent answer or an error
2. Availability - every request gets some answer (but could be old or inconsistent) - never error
3. Partition Tolerance - works despite arbitrary drops in the network (or delays)

Keys and Normal Forms:

1. SK - Super Key - combination of columns that can uniquely identify a row in the table
2. CK - Candidate Key - a Super Key that has the smallest number of columns and yet still can uniquely identify a row in the table
3. prime attribute - Any column of a table that comes from any CK
4. FD - Functional Dependence. When you have $X \rightarrow Y$ it means that if we know X, then we can figure out (or know) Y. This can take the form of multiple columns too. $XY \rightarrow RS$ means that if we know X and Y, then we can figure out both R and S. Another way to say this, is to say that R is a function of X and Y. S is also a function of X and Y.
5. 1NF - no multi-valued attributes
6. 2NF - 1NF AND no partial dependencies – No non-prime can depend on just a part of a CK.
7. 3NF - 2NF AND no transitive dependencies – For every FD of the form $X \rightarrow Y$, X is a SK or Y is prime.
8. BCNF - 3NF AND LHS of FD is always a SK. – For every FD of the form $X \rightarrow Y$, X is a SK.
9. To decompose a relation, we typically choose a FD like $X \rightarrow Y$. Replace this relation with two relations: 1) includes X and every thing it reaches. 2) includes X and everything else that isn't in 1).

Anomalies:

1. Update - could break a FD by violating it
2. Insert - adding a new row in a table could break a FD
3. Delete - we could lose some information related to a FD

Sample Questions:
Some sample relations are given below these questions. This is not an exhaustive list of possible questions. Feel free to use those to answer all of the questions multiple times. Some scenarios are also listed below. The meat of the questions will include 1-6. The rest will be sorted into smaller point short response questions.

1. Identify all super keys of the given table given the functional dependencies. (HW2)
2. Identify all the candidate keys given a list of super keys (which can be determined from above). (HW2)
3. Determine if the table is in 1NF, 2NF, 3NF or BCNF. Describe which relationship breaks the next higher order normal form. (HW2)
4. Given a table and a list of FDs, describe how to decompose the table into two new tables that are in a better normal form. (HW2)
5. Given a scenario, demonstrate an ER diagram for this situation (HW1)
6. Given an ER diagram, describe the tables (with types and keys). (in class - Access)
7. Given a scenario, which two pieces of CAP are most important?
8. Given a scenario, why is ACID important?
9. What might be the consequences if this particular feature (from DB ideas) is not part of the DBMS?

Sample Tables/FDs to practice things on (questions 1-5)

1. Table: ABCD, FD: A $\rightarrow$ B, C $\rightarrow$ D (1NF, breaks 2NF)
2. Table: ABC, FD: A $\rightarrow$ B, B $\rightarrow$ C (2NF, breaks 3NF)
3. Table: ABC, FD: AB $\rightarrow$ C, C $\rightarrow$ A (3NF, breaks BCNF)
4. Table: ABCDE, FD: AB $\rightarrow$ C, AD $\rightarrow$ E
5. Table: ABCDE, FD: AB $\rightarrow$ C, A $\rightarrow$ D
6. Table: ABCD, FD: AB $\rightarrow$ C, BC $\rightarrow$ D

Sample Scenarios to give you Ideas to practice things on (questions 7-9)

1. Libraries, Books, Lending, Customers
2. Videos, Creators, Viewers (think Youtube, TikTok)
3. Albums, Songs, Music, Playlists, Consumers (think Spotify)
4. Movies, Actors, Directors (think IMDB)
5. Images, Creators, Views, Reviews (think Google Reviews)
6. Games, Owners, Players, Current Players, Reviews, Sizes (think Steam)