**Database Literacy Test – ANSWER KEY**

1. Can a scheme be in First Normal Form and satisfy no functional dependencies? Justify your answer
2. Yes \*
3. No

First normal form has nothing to do with functional dependencies. So the intended answer is YES. Of course there is trickery answer which we have to accept:

NO, every relations satisfies \*trivial\* fds. (but only when trivial dependencies are mentioned)

1. Write SQL query “Students who take databases and also take operating systems”. Assume TAKE(Student, Class) scheme

Selfjoin (that’s the key…not 3-liner sql with just an AND)

1. With TAKE(Student, Class) as the database scheme write SQL query “Students who do not take databases”

NOT EXISTS, NOT IN, or outer join, not a 3-liner query

1. Define 2PL and state what does 2PL guarantee?

2PL is protocol in which all locks must precede all unlocks in a transaction. 2PL guarantees conflict serializability for any schedule made from 2PL transactions

1. What is normalized data?

Only atomic values allowed (first normal form), also no update anomalies and redundancies (BCNF)

(one of these two answers will be enough)

1. What is denormalized data?

Complex objects like JSON which contain sets or tuples made from sets. Join of all tables for the purpose of efficiency

(one of the two answers will be fine)

1. What does it mean for a relation to be decomposed into normal form – what two properties should decomposition have?

No update anomalies and no redundancies.

Some have written “lossless join and dependency preservation” – this should be fine too.

1. 2PL is a property of
2. Schedule
3. \*Transaction
4. Table
5. In what sense is 2PL the best we can do ?(assuming we do not know anything about transactions)

Even if one transaction violates 2PL – it is possible that non-serializable schedule can be produced (despite all other transaction satisfying 2PL)

1. NoSQL databases are particularly useful when
2. we need strict ACID properties for transactions
3. \*data is on multiple clusters
4. we make ad hoc queries
5. Functional dependency A->B means
6. One to many relationship between A and B
7. \*Many to one relationship between A and B
8. One to one relationship between A and B
9. (From 336 Challenges)

Show a database instance of Bar-Beer-Drinker database (6 tables), such that deletion of one tuple could cause deletion of all tuples in the database. What foreign keys would be needed for this and what options would have to be chosen to make a simple deletion cause such devastating effect on the database?

**[DELETED**]

ANSWER: but here is great answer provided by *Sam Sohn* from LIV section who also pointed out that none of the answers (including mine) on challenge forum was correct. Here is his answer:

For the foreign key question on the final exam:

    sells( bar)              -> bars( name)

    likes( beer)            -> beers( name)

    frequents( drinker) -> drinkers( name)

    bars( city)       -> cities( name) #NEW TABLE

    drinkers( city) -> cities( name)

    beers( manf)       -> manf( name, exportsTo) #NEW TABLE

    manf( exportsTo) -> cities( name)

This way, deleting one tuple in the cities table will delete tuples from manf(), bars(), drinkers(). Deleting from manf() deletes tuples from beers(). Deleting from beers() deletes tuples from likes(). Deleting from bars() deletes tuples from sells(). Deleting from drinkers() deletes tuples from frequents().

  Example: (ordered by precedence)

  cities: ("Piscataway")

  bars: ("Sam's Bar", "Piscataway")

  drinker: ("Sam", "Piscataway")

  manf: ("Sam's Beer Maker", "Piscataway")

  beers: ("Sam's Beer", "Sam's Beer Maker")

  sells: ("Sam's Bar", "Sam's Beer")

  likes: ("Sam", "Sam's Beer")

  frequents: ("Sam", "Sam's Bar")

1. Give an example of a database instance over the scheme ABCD which satisfies all functional dependencies

Instance with just one tuple

1. Give an example of a weak entity

Player in a team, Building in Campus etc

15 (From 336 challenges) Let’s assume our relational scheme has 5 attributes: ABCDE and the set of functions F contains all possible functional dependencies among these attributes.  
  
What is the size of the smallest minimal basis for F?  By size we mean total number of attributes in all functional dependencies (duplicates count). So for example AB->C, AD->C has size of 6 (3 attributes per fd).  In addition to the size (integer) show us the minimal basis which has this size.

A->B, B->C, C->D,D->E, E->A, size 10

1. If COUNT was not part of SQL, how many self joins would be necessary to find drinkers who like at least k beers?

k-1

(but will accept k and even k+1)

1. What is Lossless join property?

That a relation is equal to join of its projections

1. ACID - what it stands for and what each of its properties mean?

[**Atomicity**](https://en.wikipedia.org/wiki/Atomicity_(database_systems)) requires that each transaction be "all or nothing": if one part of the transaction fails, then the entire transaction fails, and the database state is left unchanged. An atomic system must guarantee atomicity in each and every situation, including power failures, errors, and crashes. To the outside world, a committed transaction appears (by its effects on the database) to be indivisible ("atomic"), and an aborted transaction does not happen.

**The**[**consistency**](https://en.wikipedia.org/wiki/Consistency_(database_systems)) property ensures that any transaction will bring the database from one valid state to another. Any data written to the database must be valid according to all defined rules, including [constraints](https://en.wikipedia.org/wiki/Integrity_constraints), [cascades](https://en.wikipedia.org/wiki/Cascading_rollback), [triggers](https://en.wikipedia.org/wiki/Database_trigger), and any combination thereof. This does not guarantee correctness of the transaction in all ways the application programmer might have wanted (that is the responsibility of application-level code) but merely that any programming errors cannot result in the violation of any defined rules.

The [**isolation**](https://en.wikipedia.org/wiki/Isolation_(database_systems)) property ensures that the concurrent execution of transactions results in a system state that would be obtained if transactions were executed serially, i.e., one after the other. Providing isolation is the main goal of [concurrency control](https://en.wikipedia.org/wiki/Concurrency_control). Depending on the concurrency control method (i.e., if it uses strict - as opposed to [relaxed](https://en.wikipedia.org/wiki/Serializability#Relaxing_serializability) - serializability), the effects of an incomplete transaction might not even be visible to another transaction.

The [**durability**](https://en.wikipedia.org/wiki/Durability_(computer_science))property ensures that once a transaction has been committed, it will remain so, even in the event of power loss, [crashes](https://en.wikipedia.org/wiki/Crash_(computing)), or errors. In a relational database, for instance, once a group of SQL statements execute, the results need to be stored permanently (even if the database crashes immediately thereafter). To defend against power loss, transactions (or their effects) must be recorded in a [non-volatile memory](https://en.wikipedia.org/wiki/Non-volatile_memory).

1. Can a table have just one superkey?   Under what conditions?

  Yes, with no functional dependencies. When key= all attributes

1. Which operations need duplicates in SQL?

aggregates

1. Which of the following is true
2. There is at least as many keys as superkeys in any relational scheme
3. \*There is at least as many superkeys as keys in any relational scheme
4. Number of keys and number of superkeys are unrelated
5. Let R(AB), S(BC) and B be a foreign key with R as referencing table and S as referenced table. Then which of the following is true

1. Natural join of R and S has the same number of tuples as S
2. \*Natural join of R and S has the same number of tuples as R
3. Natural join of R and S has more tuples than both R and S
4. Can precision be smaller than recall? Give an example of when that can be/hold or prove that it can’t.

Recall can be 100% and precision can be 1% , as small as you want, so yes. How? Say I return all documents in the database in return to query “json” as the answer. Say 1% of database documents actually contain “json”. Recall is 100% (I got all json documents) but precision is just 1% since there are only 1% of json documents among ones which I returned.

1. Can IDF of a term be zero and if so when?

Yes, when a term appears in all documents in the database. Like, say “the”

1. Someone asks for recommendation “should I use NoSQL or SQL”, provide two questions which you would ask them before you make your recommendation

Will you need ad hoc queries?

Is your data structured

Do you need strict, ACID style, concurrency control

Are you running web based high performance database application with massive number of users

Do you use large number of clusters

(and two of these or others which make sense)

1. Avoiding update anomalies is the objective of a) Query optimization, b) 2PL, c) \*Data normalization

26) What is the difference between serial and serializable schedule of transactions?

Serializable may have transactions interleaving their steps in a schedule, serializable is equivalent to serial which does not allow a transaction to start before all trasactions which started before it - finish

27) Can a relation always have lossless join decomposition into binary relations (relations with two columns)?

No, for example Student, Class, Grade would not decompose into binary tables

28) What changes more often: database scheme or database instance?

Database instance (tuples)

29) What does declarative and procedural mean in context of a programming language? (This includes query languages)

Declarative – what to do

Procedural – how to do it

30) I am using a new database system, Old Tavern DB. I have replicated table “Likes” over 5 machines and as a result I had to change my query: “How many beers does Joe like”.

Which important property is violated by Old Tavern DB?

Data Independence

31) What are assertions used for?

To assure integrity of data – to make sure database is consistent it has to satisfy assertions

32) Provide the smallest example of table which satisfies A-> B but violates AC->B and B->C

No such example exists. It is impossible. A->B entails AC->B

33. Show the “key value” representation (as JSON object) of LIKES(Drinker, Beer, Price)

[Joe, {[Heineken, $5], [Bud, $7]}

This is “pseudo JSON” – xml acceptable as well

34. Let R(AB) have 10 tuples and S(BC) have 20 tuples, what is the maximum number of tuples which natural join of R and S could have?

200

if all tuples match on the B attribute (single value)

35. Why do we add 1 in the definition of TF, and not add 1 in the definition of IDF?

Because we want zero IDF core for very frequent words which appear in ever document, but we do not want zero TF score for word which appears once in a document

36. What is CAP theorem?

In presence of partitions you cannot satisfy consistency and availability in the same time

37. Can Salary be an entity? Justify your answer

If it is just a number, no. If it is a concept – could be, as long as it has attributes…..like “how is it paid, weekly, monthly etc), is it taxed? But then it is not a number it is more like salary TYPE.

Anyway, I expect NO (and why)

If yes – provide even more intelligent explanation.

Given a query Q and a collection of documents, an IR system returns in descending order of rank 12 documents, from which only 4 were judged relevant by the user. Suppose that the entire set of documents has been evaluated and we know that there are 25 relevant documents. What are the precision and recall?

Precision = 4/12

Recall = 4/25