# Answers

**Question 1**

1. The table contains seven tuples with five attributes per tuple or row.
2. The issue lies in the fact the managers’ address attribute contains the street address, city, state and zip code all within the same attribute. This would require a statement that would have to read the MANAGER\_ADDRESS attribute and look for certain keywords for the desired city name. I would suggest breaking the MANAGER\_ADDRESS attribute up into four separate parts creating four new attributes STREET\_ADDRESS, CITY, STATE, ZIP\_CODE. This way you would be able to query the table and return results based on any of the newly created attributes.
3. Holly B. Parker and George F. Dorts both have multiple entries in the table because they are assigned to multiple projects. These redundancies not only create extra data but can create issues when querying the database. Since there are multiple entries for each of the above-named people, if you say, wanted to update the bid price for Holly B Parker and not realize that she has multiple records you might overwrite the bid price for all entries instead of just the single one you wanted to alter, creating malformed data and errors.
4. Say you want to update the project bid for PROJECT\_CODE 27-4Q. If you query based on name then you will update the bids for every project that George is currently assigned to instead of just updating 27-4Q.
5. Say you wanted to delete one project bid price from the table for Holly B. Parker. If you query based on PROJECT\_MANAGER and delete PROJECT\_BID\_PRICE based off name you will most likely delete all bids for every project she is currently assigned to instead of deleting the specific one you wanted.

**Question 2**

* Degrees of abstraction in database design
* Physical Level – The lowest level of abstraction that shows how the data is stored within the database. This level of abstraction shows the data structures in detail and assigns fields and records data types and purpose.
* Logical Level – This level describes how the data within the database are related and how they can interact with each other. Things like primary keys and foreign keys would most likely be assigned during this level to allow for the data to be interconnected and allow the data to be seen by and interact with data from other tables or databases.
* View Level – The highest level of abstraction. Sometimes users do not need to know everything about the database, rather just need certain information from it. These are called views since the user can retrieve certain information and “view” that information in a structured manner.
* Database schema and database instance
* Database Schema – This defines the overall database design, so what tables, fields, attributes etc. to be used in the database are all defined by the schema.
* Database Instance – Defines the information that is actually present in the database at any given time. Every time new information is added or information is removed this creates a new “instance” of the database since the information therein has been changed or altered in some way.

**Question 3**

table user

( first\_name text

last\_name text

email\_address text

phone\_number numeric

date\_of\_birth text )

This table would store user information pertaining to profile validation and would help in keeping people with similar information from being confused with other accounts such as first\_name, last\_name, email\_address, phone\_number, date\_of\_birth. The problem with creating a database for a social application like Facebook is the fact that people can have the same information such as names and birth dates. Collecting phone numbers and emails is an effective way of ensuring information makes its way to the proper individual profile.

table user\_activity

( posts text

liked\_posts text

friend\_requests text )

I could see a table that would track each user’s activity such as their posts, posts they liked from other users, and friend requests. This way Facebook can “fine-tune” your experience by mining this data and tailoring each person’s feed and ad experience to things that pertain to them on a personal level.

table personalization

(keywords text

ad\_platforms text

interests text)

I could see a table in where Facebook keeps track of things you mention in posts or comments then they can base ad experiences and other feed material based off of this information, keeping your page relevant and interesting to you on a personal level.

**Question 4**

*App #1 – Golf GPS and Scorecard*

Table User –

|  |  |  |
| --- | --- | --- |
| USER\_ID(PK) | USERNAME | EMAIL |
| 01 | Gabriel Wallace | Foo.bar@gmail.com |
| 02 | Jim Lahey | Jim.lahey@sunnyvale.net |
| 03 | Luke Skywalker | WompRatSniper@tatooine.org |

Table Courses –

|  |  |  |
| --- | --- | --- |
| COURSE\_ID(PK) | COURSE\_NAME | COURSE\_PAR |
| 01-A | Thompson Oaks | 27 |
| 02-A | Inver Hills Golf Course | 72 |
| 03-A | Mendota Heights Par 3 | 28 |

Table clubs

|  |  |  |
| --- | --- | --- |
| CLUB\_ID(PK) | CLUB\_NAME | AVERAGE\_DIST |
| 01 | Driver | 185 |
| 02 | 3 Wood | 150 |
| 03 | 5 Wood | 130 |

*App #2 – Lyft*

Table rider

|  |  |  |
| --- | --- | --- |
| RIDER\_ID(PK) | USERNAME | EMAIL |
| 01-01R | Gabriel Wallace | [Foo.Bar@gmail.com](mailto:Foo.Bar@gmail.com) |
| 01-02R | George Costanza | [GCostanza@aol.com](mailto:GCostanza@aol.com) |
| 01-03R | Snake Plissken | SolidSnake@metalgear.com |

Table driver

|  |  |  |
| --- | --- | --- |
| DRIVER\_ID(PK) | PHONE | RATING |
| 01-01D | (555)555-8976 | 5 |
| 01-02D | (555)555-7183 | 4 |
| 01-03D | (555)555-9254 | 4 |

Table rides

|  |  |  |
| --- | --- | --- |
| RIDER\_ID (FK) | RIDE\_ID | DISTANCE |
| 01-01R | R123f | 12 Miles |
| 01-02R | R432f | 10 Miles |
| 01-03R | R437f | 15 miles |

*App #3 – Spotify*

Table user

|  |  |  |
| --- | --- | --- |
| *USER\_ID(PK)* | *USERNAME* | *EMAIL* |
| *0101* | *SylvesterSneer* | [SSneer@gmail.com](mailto:SSneer@gmail.com) |
| *0102* | *FooBar* | [Foo.Bar@gmail.com](mailto:Foo.Bar@gmail.com) |
| *0103* | *DarthVader* | *DVader@sithnet.com* |

Table user\_info

|  |  |  |
| --- | --- | --- |
| USER\_ID(FK) | PLAYLIST\_COUNT | SONG\_COUNT |
| 0101 | 5 | 257 |
| 0102 | 3 | 5643 |
| 0103 | 6 | 654 |

Table artist

|  |  |  |
| --- | --- | --- |
| ARTIST\_ID(PK) | ARTIST\_NAME | TOTAL\_TRACKS |
| 0001 | GRiZ | 45 |
| 0002 | Pretty Lights | 80 |
| 0003 | Big Gigantic | 32 |