

Oracle SQL  
Lab Assignment 1  
Wolfgang C. Strack  
Student ID#: \*\*\*\*7355  
Due date: 3 July 2015  
Date handed in: 3 July 2015

**1. What is an entity? How is it defined in a database? What does a row and column represent in a table?**

An entity is something that can be classified as a single object and that object can be a person, place, or anything that which data about it can be classified and stored. An entity is usually defined in a database as a table. The entity's classified characteristics, which are more commonly referred to as attributes, represent columns in a table. Finally, the rows in a table represent separate, unique records that all fall under that specific entity.

**2. What are primary and foreign keys? Explain in detail. What are the implications of using these keys in different tables?**

Primary keys are represented in tables as a field or group of fields that are not null and uniquely identify all individual records within that table, by definition. For example, a primary key can be classified in an entity such as a student such that the primary key would be every student's unique student ID. Since all students have a different, unique student ID, it can be considered a primary key if we let "student" be the entity in our database. A foreign key on the other hand can be considered a secondary key. To be exact, what makes a foreign key "foreign" is that it comes from a different table from the table it is stored in. These two tables, the table in which the key is considered foreign and the table from which the key comes from, share some sort of relationship that allows for them to share unique keys that correspond to their records. The implications of using these keys in different tables is, just as mentioned before, that there is a relationship between the records in the different tables which share keys. By sharing the foreign key, that table can share information to the other table that is using the foreign key.

**3. What is referential integrity? Describe four advantages of referential integrity?**

Referential integrity is a concept defined in database systems which enforces consistency in the relationships between tables. For example, when there is any type of update to a table that shares relationships (i.e. foreign keys) with other tables—such as an insert, update, or delete—it is always checked with links, or the tables that the updated table's foreign keys come from.

Four advantages of referential integrity include helping to normalize data by spreading out attributes across multiple tables instead of using just a single large table. Another advantage is implied from this, that data is maintained consistently across the multiple tables; there is less of a chance for error if the tables are properly linked with referential integrity. Further advantages can be implied from that as well; a third advantage is the productivity that comes with properly linked tables under referential integrity. By updating one table, all tables that are linked to the updated table will also be updated with whatever information they share. In other words, one does not have to change multiple tables if they want to change the one shared value. Finally, a fourth advantage can then be subsequently derived from the third which is that referential integrity is declared and therefore automatic rather than being manually programmed. In other words, one does not have to manually program the rules for linking functionality between tables as it is automatic through referential integrity rules.

## **5. What is data dictionary? What is it used for?**

In a database, the data dictionary is the information about the database. Represented as a read-only set of tables, the data dictionary is used by both the database and the database administrators/users to understand the architecture of the database. For example, the data dictionary contains information about how the database's tables are structured, including things like column names, default values of columns, and other information about the tables' fields.

## **6. What are the three rules of normalization? How is it applied in creating a DB Schema?**

The three rules of normalization, otherwise known as the three normal forms in structuring a table, are:

1. there must be no repeating of the same attribute or group of attributes in a row,
2. after qualifying for the first normal form, data in the table must be wholly dependent on the primary key's value such that the unnecessary duplication of data in individual rows can be avoided,
3. and after qualifying for the first two normal forms, the third normal form requires that data in each column, besides keys, must not be interdependent such that the data could already be defined somewhere else or can just be calculated from existing values.