**Chapter 3 - Review Questions**

**3. Why are entity integrity and referential integrity important in a database?**

* Entity integrity is important because it means that a proper search for an existing tuple (row) will always be successful. And the failure to find a match on a row search will always mean that the row for which the search is conducted does not exist in that table. Referential integrity is important, because its existence ensures that it will be impossible to assign a non-existing foreign key value to a table. For example, the referential integrity enforcement in a SALESREP is assigned to CUSTOMER relationship means that it will be possible for a customer not have a sales rep assigned to him or her, but it will be impossible to assign non-existing sales rep to a customer.

**5. Use the small database shown in Figure Q3.5 to illustrate the difference between a natural join, an equijoin, and an outer join.**

* A natural join links the two tables with the PRODUCT of the two as shown:

|  |  |  |  |
| --- | --- | --- | --- |
| **STU\_CODE** | **PROF\_CODE** | **PROF\_CODE** | **DEPT\_CODE** |
| 100278 |  | 1 | 2 |
| 128569 | 2 | 1 | 2 |
| 512272 | 4 | 1 | 2 |
| 531235 | 2 | 1 | 2 |
| 531268 |  | 1 | 2 |
| 553427 | 1 | 1 | 2 |
| 100278 |  | 2 | 6 |
| 128569 | 2 | 2 | 6 |
| 512272 | 4 | 2 | 6 |
| 531235 | 2 | 2 | 6 |
| 531268 |  | 2 | 6 |
| 553427 | 1 | 2 | 6 |
| 100278 |  | 3 | 6 |
| 128569 | 2 | 3 | 6 |
| 512272 | 4 | 3 | 6 |
| 531235 | 2 | 3 | 6 |
| 531268 |  | 3 | 6 |
| 553427 | 1 | 3 | 6 |
| 100278 |  | 4 | 4 |
| 128569 | 2 | 4 | 4 |
| 512272 | 4 | 4 | 4 |
| 531235 | 2 | 4 | 4 |
| 531268 |  | 4 | 4 |
| 553427 | 1 | 4 | 4 |

Next, a SELECT is performed on the PRODUCT generated in the first step to yield only the rows for which the PROF\_CODE values in the STUDENT table are matched in the PROF table.

|  |  |  |  |
| --- | --- | --- | --- |
| **STU\_CODE** | **PROF\_CODE** | **PROF\_CODE** | **DEPT\_CODE** |
| 128569 | 2 | 2 | 6 |
| 512272 | 4 | 4 | 4 |
| 531235 | 2 | 2 | 6 |
| 553427 | 1 | 1 | 2 |

Finally, a PROJECT is performed to produce the natural join output by listing only a single copy of each attribute, thereby eliminating duplicate columns. The final outcome of a natural join yields a table that does not include unmatched pairs and provides only the copies of the matches as shown:

|  |  |  |
| --- | --- | --- |
| **STU\_CODE** | **PROF\_CODE** | **DEPT\_CODE** |
| 128569 | 2 | 6 |
| 512272 | 4 | 4 |
| 531235 | 2 | 6 |
| 553427 | 1 | 2 |

* An equijoin’s results depend on the specified conditions. For example, if the equijoin specifies to show the STU\_CODE for whom the PROF\_CODE is 2, then the output will be:

|  |  |  |
| --- | --- | --- |
| **STU\_CODE** | **PROF\_CODE** | **DEPT\_CODE** |
| 128569 | 2 | 6 |
| 531235 | 2 | 6 |

* In the outer join, the unmatched pairs would be retained and the values that do not have a match in the other table would be left null. The output will be:

|  |  |  |
| --- | --- | --- |
| **STU\_CODE** | **PROF\_CODE** | **DEPT\_CODE** |
| 100278 |  |  |
| 128569 | 2 | 6 |
| 512272 | 4 | 4 |
| 531235 | 2 | 6 |
| 531268 |  |  |
| 553427 | 1 | 2 |
|  | 3 | 6 |

**9. What are homonyms and synonyms, and why should they be avoided in database design?**

* Homonyms appear when more than one attribute has the same name. Synonyms exist when the same attribute has more than one name. It is crucial to avoid both to avoid inconsistencies. Synonyms make it difficult to keep track of foreign keys if they are named differently from the primary keys they point to. They should be avoided in database design so the data dictionary is not cumbersome to use and most importantly to avoid inconsistencies.

**16. Suppose you wanted quick lookup capability to get a listing of all plays directed by a given director. Which table would be the basis for the INDEX table, and what would be the index key?**

* The PLAY table would be the basis for the appropriate INDEX table. The index key would be the attribute DIR\_NUM.

**Chapter 3 – Problems**

**4. Do the tables exhibit entity integrity? Answer yes or no, and then explain your answer.**

|  |  |  |
| --- | --- | --- |
| **TABLE** | **ENTITY INTEGRITY** | **EXPLANATION** |
| EMPLOYEE | Yes | Each EMP\_CODE value is unique and there are no nulls. |
| BENEFIT | Yes | Each ***combination*** of EMP\_CODE and PLAN\_CODE values is unique and there are no nulls. |
| JOB | Yes | Each JOB\_CODE value is unique and there are no nulls. |
| PLAN | Yes | Each PLAN\_CODE value is unique and there are no nulls. |

**5. Do the tables exhibit referential integrity? Answer yes or no, and then explain your answer. Write NA (Not Applicable) if the table does not have a foreign key.**

|  |  |  |
| --- | --- | --- |
| **TABLE** | **REFERENTIAL INTEGRITY** | **EXPLANATION** |
| EMPLOYEE | Yes | Each JOB\_CODE value in EMPLOYEE points to an existing JOB\_CODE value in JOB. |
| BENEFIT | Yes | Each EMP\_CODE value in BENEFIT points to an existing EMP\_CODE value in EMPLOYEE and each PLAN\_CODE value in BENEFIT points to an existing PLAN\_CODE value in PLAN. |
| JOB | NA |  |
| PLAN | NA |  |