**1.10)** COURSE and SECTION have a relationship shared by the Course\_number. COURSE and PREREQUISITE have a relationship shared by Course\_number, where PREREQUISITE uses Course\_number, from COURSE, in both of its columns. STUDENT and GRADE\_REPORT have a relationship shared by the Student\_number. SECTION and GRADE\_REPORT have a relationship shared by the Section\_identifier.

**1.12)** Simple: The Course\_name data item within each COURSE must be no longer than 10 alphabetic characters. The Student\_number data item within each STUDENT record must be 4 digit characters.

Referential: Every PREREQUISITE record must be related to a COURSE record. Every SECTION record must be related to a COURSE record.

Key/Uniqueness: Every COURSE record must have a unique value for Course\_number. Every STUDENT record must have a unique value for Student\_number.

**1.14)** a) In STUDENT column Major, in COURSE columns Course\_number and Department, in SECTION column Course\_number, and in PREREQUISITE columns Course\_number and Prerequisite\_number.

b) Yes.

|  |  |  |  |
| --- | --- | --- | --- |
| Course\_name | Department | Course\_number | Credit\_hours |
| Intro to Computer Science | CS | 1310 | 4 |
| Data Structures | CS | 3320 | 4 |
| Discrete Mathematics | MATH | 2410 | 3 |
| Database | CS | 3380 | 3 |

**COURSE**

**SECTION**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Section\_identifier | Course\_number | Semester | Year | Instructor |
| 85 | 2410 | Fall | 07 | King |
| 92 | 1310 | Fall | 07 | Anderson |
| 102 | 3320 | Spring | 08 | Knuth |
| 112 | 2410 | Fall | 08 | Chang |
| 119 | 1310 | Fall | 08 | Anderson |
| 135 | 3380 | Fall | 08 | Stone |

|  |  |
| --- | --- |
| Course\_number | Prerequisite\_number |
| 3380 | 3320 |
| 3380 | 2410 |
| 3320 | 1310 |

**PREREQUISITE**

This restructure only works with the integrity constraint that no Course\_number can be the same regardless of the Department.

**2.14)** A) Three-Tier Client/Server Architecture

B) The three-tier architecture is good for web based systems because the middle tier would be good to hold any constraints or procedures when it comes to making an airline reservation or sell airline tickets. For example, if you must be 25 to make a reservation, the middle tier would check the age before even trying to access any of the open airline reservations. If the age is old enough then the middle tier accesses the data from the database server for any open reservations.

C) Centralized Architecture wouldn’t be good because there is a capacity restraint and there are normally thousands and thousands of flights a day. Basic Client/Server Architecture isn’t a good choice because you must have dedicated client machines. Two-Tier Client/Server Architecture isn’t a good choice because is good for searching, such as finding dictionary definitions or books in a library catalog. The Two-Tier Client/Server Architecture could work but the Three-Tier is the best option because it has that buffer/middle tier to make restrictions with business rules.

**3.11)** (a) No integrity constraints violated.

(b) Referential integrity constraint is violated because Dnum ‘2’ doesn’t refer to any Dnumber in DEPT\_LOCATIONS.

(c) Unique key integrity constraint is violated because Dnumber is the only primary key making it the unique key and you can’t have identical Dnumbers (‘4’).

(d) Entity integrity constraint violated because Pno is a primary key in WORKS\_ON can’t be null.

(e) No integrity constraints violated.

(f) No integrity constraints violated.

(g) Referential integrity constraint violated because the EMPLOYEE tuple being removed is referenced by tuples from EMPLOYEE, DEPARTMENT, WORKS\_ON, and DEPENDENT.

(h) Referential integrity constraint violated because the PROJECT tuple is being referenced by tuples in WORKS\_ON.

(i) No integrity constraints violated.

(j) No integrity constraints violated because we inserted in (a).

(k) No integrity constraints violated.

To enforce these constraints, you can reject the command, cascade delete (deletes everything related to the tuple), or cascade update (updates everything related to the tuple). These are just some of the few was to enforce the constraints.

**3.16)** Assume STUDENT SSN is unique and not NULL

Assume Course# is unique and not NULL.

Assume Book\_ISBN is unique and not NULL.

In ENROLL SSN, Course#, and Quarter are FK. In BOOK\_ADOPTION Course#, Quarter and Book\_ISBN are FK.