**Answers to Problems and Exercises**

1. StudentID in STUDENT because it is a primary key and the index would enforce uniqueness of the key; also, StudentID in STUDENT and in REGISTRATION is used in a WHERE clause for joining the STUDENT and REGISTRATION tables, so it likely makes sense to create an index on StudentID in REGISTRATION as well.

GPA in STUDENT because it is a nonkey cluster attribute used to qualify record retrieval

StudentName in STUDENT because it is a nonkey attribute used to sort records

StudentID, CourseID in REGISTRATION because it is a concatenated primary key and the index would enforce uniqueness of the key

1. CREATE UNIQUE INDEX STUPKINDX ON STUDENT (StudentID);

CREATE INDEX STUDREGINDX ON REGISTRATION (StudentID);

CREATE INDEX CLUST\_INDX

ON STUDENT (GPA)

CLUSTER;

CREATE INDEX NAMEINDX ON STUDENT (StudentName);

CREATE UNIQUE INDEX REGSPKINDX

ON REGISTRATION (StudentID, CourseID);

1. How will numeric value 3,456,349.2334 be stored assuming various Oracle data types?

|  |  |  |
| --- | --- | --- |
|  | Oracle data type | Stored Value |
| a. | NUMBER(11) | 3456349 |
| b. | NUMBER(11,1) | 3456349.2 |
| c. | NUMBER(11,-2) | 3456300 |
| d. | NUMBER(6) | Not accepted; exceeds precision |
| e. | NUMBER | 3456349.2334 |

1. Recommendations for denormalization:

EMPLOYEE SCHEDULE

(DepartmentID, EmployeeID, WhereWork, EmployeeName, EmployeeAddress, Date)

*A many-to-many relationship (associative entity) with nonkey attributes*: Rather than joining three files to extract data from the two basic entities in the relationship, it may be advisable to combine attributes from one of the entities into the record representing the relation in the many-to-many relationship, thus avoiding one join in many data access modules. This approach is advantageous as this joining will occur frequently.

DEPARTMENT

(DepartmentID, ManagerID, SalesGoal, StoreID, Region, ManagerID, SquareFeet)

This *reference data* denormalization option wouldn't be recommended since the table STORE is further related to a table MANAGER, and there are probably more than just a few departments in each STORE.