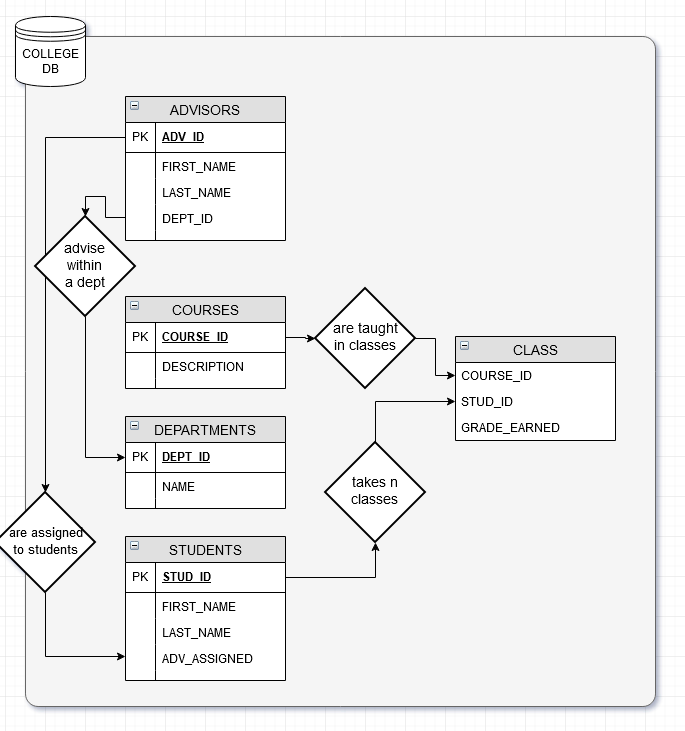
1. *What is an entity?* 
   1. An entity is data that represents nouns like Customers, Orders, Storehouses, ect.
2. *What is an attribute?* 
   1. An attribute is data that describes the above entities, like CUSTOMER\_NAME, ORDER\_NUMBER, STOREHOUSE\_ADDRESS, ect.
3. *What is a relationship? What is a one-to-many relationship?* 
   1. A database relationship is a logical association of common attributes amongst entities
   2. A 1 to many relationship can be visualized as a Doctor-Patient relationship. A doctor may have many patients under their care
4. *What is a repeating group?* 
   1. A repeating group is more than 1 attribute value assigned to an entity occupying the same row. This is an unnormalized relationship
5. *What is a relation?* 
   1. A relation is a n-dimensional table of associated data, wherein the related data is connected by a common field
6. *What is a relational database?*
   1. A relational database is a collection of n-dimensional tables
7. *Describe the shorthand representation of the structure of a relational database. Illustrate this technique by representing the database for Colonial Adventure Tours as shown in Figures 1-4 through 1-8 in Chapter 1.*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **GUIDE** |  |  |  |  |  |  |  |  |
| GUIDE\_NUM, | LAST\_NAME, | FIRST\_NAME, | ADDRESS, | CITY, | STATE, | POSTAL\_CODE | PHONE\_NUM | HIRE\_DATE |
|  |  |  |  |  |  |  |  |  |
| **TRIP** |  |  |  |  |  |  |  |  |
| TRIP\_ID | TRIP\_NAME | START\_LOCATION | STATE | DISTANCE | MAX\_GRP\_SIZE | TYPE | SEASON |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| **CUSTOMER** |  |  |  |  |  |  |  |  |
| CUST\_NUM | LAST\_NAME | FIRST\_NAME | ADDRESS | CITY | STATE | POSTAL\_CODE | PHONE\_NUM |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| **RESERVATION** |  |  |  |  |  |  |  |  |
| RESERVATION\_ID | TRIP\_ID | TRIP\_DATE | NUM\_PERSONS | TRIP\_PRICE | OTHER\_FEES | CUST\_NUM |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| **TRIP\_GUIDES** |  |  |  |  |  |  |  |  |
| TRIP\_ID | GUIDE\_NUM |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

* 1. CUSTOMERS are related to RESERVATIONS. RESERVATIONS are related to TRIP\_IDs, TRIP\_IDs can reference GUIDE\_NUMs from the TRIP\_GUIDES table or more trip information from the TRIP table.

1. *How do you qualify the name of a field, and when do you need to do this?*
   1. Qualifying a field means to directly call it with it’s table name. TRIP.TRIP\_ID and TRIP\_GUIDES.TRIP\_ID is an example.
2. *What does it mean for a column to be functionally dependent on another column?*
   1. A field is functionally dependent on another when the values of that field directly affect the values of the other.
3. *What is a primary key? What is the primary key for each of the tables in the Colonial Adventure Tours database shown in Chapter 1?*
   1. Primary keys are uniquely identifiable information associated to a record and which all other fields are functionally dependent on. This can be one or more fields
   2. GUIDE PK = GUIDE\_NUM, TRIP PK = TRIP\_ID, CUSTOMER PK = CUST\_NUM, RESERVATION PK = RESERVATION\_ID
4. *A database at a college must support the following requirements:* 
   * *For a department, store its number and name.*
   * *For an advisor, store his or her number, last name, first name, and the departmentnumber to which the advisor is assigned.*
   * *For a course, store its code and description (for example, MTH110, Algebra).*
   * *For a student, store his or her number, first name, and last name. For each coursethe student takes, store the course code, course description, and grade earned. Also, store the number and name of the student’s advisor. Assume that an advisor might advise any number of students but that each student has just one advisor.*
   * *Design the database for the preceding set of requirements. Use your own experience as a student to determine any functional dependencies. List the tables, columns, and relationships. In addition, represent your design with an E-R diagram.*
5. *Define first normal form.*
   1. 1NF is achieved when the table has no repeating groups, all values are of the same type, each field has a unique name, and the order in which data is stored doesn’t change the way queries fetch the data.
6. *Define second normal form. What types of problems might you encounter using tables that are not in second normal form?*
   1. 2NF is achieved when the table is in 1NF and there are no partial dependencies (ie fields are organized in tables such that it contains only directly relevant attributes to that entity)
7. *Define third normal form. What types of problems might you encounter using tables that are not in third normal form?*
   1. 3NF is achieved when the table is in 2NF and there are no transitive dependencies (ie fields that would depend on a non-primary/composite key attribute are separated to their own table with a PK that could be referenced from the original table they were removed from)
8. *Using the functional dependencies you determined in Question 11, convert the following table to an equivalent collection of tables that are in third normal form.STUDENT*

*( STUDENT\_NUM,*

*STUDENT\_LAST\_NAME,*

*STUDENT\_FIRST\_NAME,ADVISOR\_NUM,*

*ADVISOR\_LAST\_NAME,*

*ADVISOR\_FIRST\_NAME,*

*(COURSE\_CODE, DESCRIPTION, GRADE)*

*)*

1. STUDENTS

( STUDENT\_ID,

STUDENT\_LAST\_NAME,

STUDENT\_FIRST\_NAME

)

ADVISORS

( ADVISOR\_ID,

ADVISOR\_LAST\_NAME,

ADVISOR\_FIRST\_NAME,

COURSE\_ID\_ASSIGNED

)

COURSES

( COURSE\_ID,

DESCRIPTION

)

CLASSES

( COURSE\_ID,

STUDENT\_ID,

GRADE

)

This could alternatively be done by eliminating the CLASSES table and creating a composite-primary key between COURSE\_ID AND STUDENT\_ID in the COURSES table

COURSES

( **COURSE\_ID**, // COMPOSITE-

**STUDENT\_ID**, // PRIMARY KEY

DESCRIPTION,

GRADE // however, this creates a partial dependency between

) // DESCRIPTION and COURSE\_ID.