**1. Define the following terms and then create an example.** **(3 points each)**

**a. Super key:** A super key k is an/are attribute(s) which is/are enough to identify a tuple t uniquely in a relation r.

Example: Student ID in students’ relation is a super key whereas the student name is not a super key as there will be more than one student with the same name.

**b. Candidate key:** A candidate key is any super key that is minimal.

Example: Student ID is a super key. Student name and department together is a candidate key for student relation.

**c. Primary key:** One of the candidate keys is selected to be primary key. We must choose one that changes rarely.

Example: Student ID can be a primary key with which a student is identified.

**d. Foreign key:** Value in one relation must appear as primary key in another relation.

Example: The room no. tuple is a foreign key, as it is available as primary key in section relation and as foreign key in classroom relation.

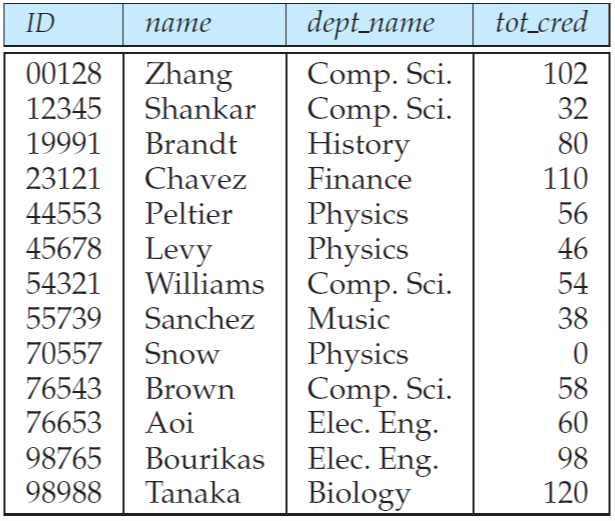
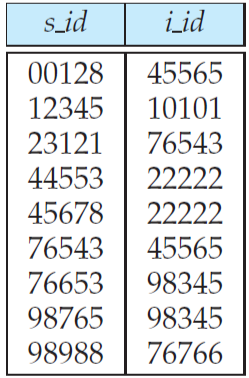
**2. Look at the time\_slot relation. You can find this relation on page 1271 of the textbook. In this relation the primary key is made up of 3 attributes. The only attribute not in the key is end\_time. Try to tell me why this is the case.** **(2 points)**

The attributes time\_slot\_id, day and start\_time may be different when it includes id and day. If start\_time and end\_time are same for some courses it may end up in a clash between classes.

**3. What is the result of the following compound relational algebra statement? Make sure you base it on the textbook data (pages 1276 – 1281) and show your work (using the textbook data) not just the answer. In other words, show each table of the original data, then the intermediate table(s) in the order they occur, and finally the resulting table. (5 points)**

1. **σs\_id=ID (student x advisor)**

**R1. Student Relation R2. Advisor relation**

** **

**Output result set:** Students with advisors are shown below whereas students who do not have an advisor will not appear.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **name** | **dept name** | **tot cred** | **s\_id** | **i\_id** |
| 00128 | Zhang | Comp. Sci. | 102 | 00128 | 45565 |
| 12345 | Shankar | Comp. Sci. | 32 | 12345 | 10101 |
| 23121 | Chavez | Finance | 110 | 23121 | 76543 |
| 44553 | Peltier | Physics | 56 | 44553 | 22222 |
| 45678 | Levy | Physics | 46 | 45678 | 22222 |
| 76543 | Brown | Comp. Sci. | 58 | 76543 | 45565 |
| 76653 | Aoi | Elec. Eng. | 60 | 76653 | 98345 |
| 98765 | Bourikas | Elec. Eng. | 98 | 98765 | 98345 |
| 98988 | Tanaka | Biology | 120 | 98988 | 76766 |

**4. Consider the following expressions, which use the result of a relational algebra operation as the input to another operation. For each expression, explain in words what the expression does.**

**a. σ year≥2009(takes) X** **student**

The result set will be a new relation with attributes of **Student** (takes at least one course) relation and **Takes** relation by a natural join in the year 2009 and after 2009.

**b. σ year≥2009(takes X student)**

The result set will be a new relation with attributes of **Student** (takes at least one course) relation and **Takes** relation by a natural join in the year 2009 and after 2009. Same result as **4a.**

**c. π ID, name, course\_id (student X takes)**

The result set will be a list of ID, Name, Course\_id of all students who will be taking any course.

**5. Consider the relational database of Figure 2.14. Give an expression in the relational algebra to express each of the following queries:**

**a. Find the names of all employees who live in city “Miami”.**

Π name (σ city = “Miami” (employee))

**b. Find the names of all employees whose salary is greater than $100,000.**

Π name (σ salary = $100,000 (employee))

**c. Find the names of all employees who live in “Miami” and whose salary is greater than $100,000.**

Π name (σ city = “Miami” ∩ salary > $100,000 (employee))

**6. Consider the bank database of Figure 2.15. Give an expression in the relational**

**algebra for each of the following queries.**

**a. Find the names of all branches located in “Chicago”.**

Π branch\_name (σ branch\_city = “Chicago” (branch))

**b. Find the names of all borrowers who have a loan in branch “Downtown”.**

Π customer\_name (σ branch\_name = “Downtown” (borrower X loan))

**7. Consider the advisor relation shown in Figure 2.8, with s\_id as the primary key of advisor. Suppose a student can have more than one advisor. Then, would s\_id still be a primary key of the advisor relation? If not, what should the primary key of advisor be?**

No, s\_id cannot be a primary key. There will be two tuples(rows) for each student as a student can have more than one advisor. There will be two primary keys s\_id and i\_id from advisor relation.

**8. Consider the relational database of Figure 2.14. Give an expression in the relational algebra to express each of the following queries:**

**a. Find the names of all employees who work for “First Bank Corporation”.**

Π person\_name (σ company\_name = “First Bank Corporation” (works))

**b. Find the names and cities of residence of all employees who work for “First Bank Corporation”.**

Π person\_name, city (employee X (σ company\_name = “First Bank Corporation” (works)))

**c. Find the names, street address, and cities of residence of all employees who work for “First Bank Corporation” and earn more than $10,000?**

Π person\_name, street, city (σ company\_name = “First Bank Corporation” ∩ salary > $10000) (works X employee))

**9. Consider the bank database of Figure 2.15. Give an expression in the relational**

**algebra for each of the following queries:**

**a. Find all loan numbers with a loan value greater than $10,000.**

Π loan\_number (σ amount > $10000 (loan))

**b. Find the names of all depositors who have an account with a value greater than $6,000.**

Πcustomer\_name (σ balance > $6000(depositor X account))

**c. Find the names of all depositors who have an account with a value greater than $6,000 at the “Uptown” branch.**

Π customer\_number (σ balance > $6000 ∩ branch\_name=” Uptown” (depositor X account))