**Lab 2**

1.  Use the sample of a database shown below to work Problems 1.1 through 1.5.  
This database has 3 relations.

**1.1[1]** For each of the tables in the database, identify super keys, candidate keys, primary key and the foreign keys. If a table does not have a foreign key, write NONE.

Ans :-

Assume every student has a unique phone number

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Relations | Super key | Candidate key | Primary key | Foreign key |
| Student | (STD\_ID),(PHONE),(STD\_ID,SF\_NAME),  (STD\_ID,SLNAME)(STD\_ID,PHONE)  (STD\_ID,MAJOR) | (STD\_ID)  (PHONE) | STD\_ID | NONE |
| Course | (C\_CODE),(CRS\_NAME)  (C\_CODE,CRDT)(CRS\_NAME,CRDT) | (C\_CODE)  (CRS\_NAME) | C\_CODE | NONE |
| Grade | (CRS\_NAME,SL\_NAME) | (CRS\_NAME,SL\_NAME) | (CRS\_NAME,  SL\_NAME) | CRS\_NAME  SL\_NAME |

**1.2 (1)** Do the tables exhibit entity integrity? Answer Yes or No, then justify your answer.

ANS:

YES, because none of the primary key in the relations is null.

**1.3(1)** Do the tables exhibit referential integrity? Answer Yes or No, then justify your answer.

ANS:

YES, because every foreign key value match a tuple in their home relation.

**1.4(1)** Comment on each table. Can you propose a better organization of data? Justify your answer.

ANS:

* Student Relation looks fine.
* Course Relation violates the Relation property that each cell should have one

atomic value but the Qualified faculty column contains more than one value.

a better solution would be to have separate column for each qualified faculty.

* Grade relation: would have been better if it had Course code and student id as the

foreign keys.

**1.5(1)** For each of the tables in the database, create two new rows such that first one violates entity integrity and the second one violates referential integrity. If such a row does not exist, write NONE.

ANS:

                        STUDENT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SDT\_ID | SF\_NAME | SLNAME | PHONE | MAJOR |
| 935499 | Jill | Meek | (345)345-5216 | CSC |
| 355869 | Cathy | Liu | (356)467-7488 | PHY |
| 577885 | James | Smith | (456)435-4658 | CHM |
| 127345 | Joy | Brown | (108)365-8976 | CSC |
| 456328 | Lisa | Williams | (239)879-3723 | ENG |
| 732489 | Cathy | Cheng | (213)895-4367 | MUS |
| null | zelalem | zergaw | (706)689-6985 | CSC |
| NONE |  |  |  |  |

 COURSE

|  |  |  |  |
| --- | --- | --- | --- |
| C\_CODE | CRS\_NAME | CRDT | QUALIFIED\_FACULTY |
| PHY304 | Relativity | 3 | Wolfe, Lathrope, Macy |
| CHM208 | Organic Chemistry | 4 | Walker, Bosch |
| CHM209 | Physical Chemistry | 4 | Walker, Shara |
| MUS338 | Jazz Ensemble | 3 | Gross |
| CSC121 | Programming | 3 | LeMack, Kurup, Naik |
| ENG345 | Creative Writing | 3 | Hanna, Cooley |
| CSC122 | Organization | 3 | LeMack, Kurup |
| CSC124 | Architecture | 3 | Kurup, Naik, Ray |
| PHY207 | Dynamics | 4 | Wolfe, Lathrope, Levy |
| null | Database | 4 | Walker, Shara |
| NONE |  |  |  |

  GRADE

|  |  |  |  |
| --- | --- | --- | --- |
| CRS\_NAME | SL\_NAME | GRADE | FACULTY |
| Creative Writing | Cheng | B | Cooley |
| Dynamics | Brown | C | Lathrope |
| Dynamics | Smith | D | Wolfe |
| Programming | Brown | C | Kurup |
| Relativity | Cheng | A | Wolfe |
| Relativity | Meek | B | Wolfe |
| Jazz Ensemble | Williams | A | Gross |
| Organic Chemistry | Williams | C | Walker |
| NONE |  |  |  |
| Software engineering | Williams | A | Michael |

**2. (1)** Find union, intersection and difference of the following two relations.   
Further, find **∏A,C (σ(A=a2) ˄ (B=b2)(R)).**

|  |  |  |
| --- | --- | --- |
| A | B | C |
| a1 | b1 | c1 |
| a1 | b1 | c2 |
| a2 | b2 | c1 |
| a2 | b1 | c2 |

|  |  |  |
| --- | --- | --- |
| A | B | C |
| a1 | b2 | c2 |
| a2 | b2 | c1 |
| a1 | b1 | c2 |

R S

**ANS:**

**Union R ∪ S**

|  |  |  |
| --- | --- | --- |
| A | B | C |
| a1 | b1 | c1 |
| a1 | b1 | c2 |
| a2 | b2 | c1 |
| a2 | b1 | c2 |
| a1 | b2 | c2 |

**Intersection R ∩ S**

|  |  |  |
| --- | --- | --- |
| A | B | C |
| a1 | b1 | c2 |
| a2 | b2 | c1 |

**Difference R – S**

|  |  |  |
| --- | --- | --- |
| A | B | C |
| a1 | b1 | c1 |
| a2 | b1 | c2 |

**∏A, C (σ(A=a2) ˄ (B=b2)(R)).**

|  |  |
| --- | --- |
| A | C |
| a2 | c1 |

**3.(1)**Find equi-join, natural join, left outer join and right outer join.

|  |  |  |
| --- | --- | --- |
| D | C | E |
| d1 | c1 | e2 |
| d2 |  | e3 |
| d3 | c2 | e2 |
| d4 | c2 | e1 |

R S

|  |  |  |
| --- | --- | --- |
| A | B | C |
| a1 | b1 | c1 |
| a2 | b1 | c2 |
| a3 | b2 | c1 |
| a4 | b1 | c3 |

ANS:

**R ⋈R.c=S.c S (equi-join)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | B | R.C | D | S.C | E |
| a1 | b1 | c1 | d1 | c1 | e2 |
| a2 | b1 | c2 | d3 | c2 | e2 |
| a2 | b1 | c2 | d4 | c2 | e1 |
| a3 | b2 | c1 | d1 | c1 | e2 |

**R ⋈ S (Natural-join)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | D | E |
| a1 | b1 | c1 | d1 | e2 |
| a2 | b1 | c2 | d3 | e2 |
| a2 | b1 | c2 | d4 | e1 |
| a3 | b2 | c1 | d1 | e2 |

**R ⋊ S (Left -outer join)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | D | E |
| a1 | b1 | c1 | d1 | e2 |
| a2 | b1 | c2 | d3 | e2 |
| a2 | b1 | c2 | d4 | e1 |
| a3 | b2 | c1 | d1 | e2 |
| a4 | b1 | c3 | Null | null |

**R ⋉ S (right -outer join)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | D | E |
| a1 | b1 | c1 | d1 | e2 |
| a2 | b1 | c2 | d3 | e2 |
| a2 | b1 | c2 | d4 | e1 |
| Null | Null |  | d2 | e3 |

**4.(1.5)** A relational database contains details about journeys from Chicago to a variety of destinations and contains the following relations:

Operator (opCode, opName)

Journey (opCode, destCode, price)

Destination (destCode, destName, distance)

* Each operator is assigned a unique code (opCode) and the relation Operator records the association between this code and the Operator’s name (opName).
* Each destination has a unique code (destCode) and the relation Destination records the association between this code and the destination name (destName), and the distance of the destination from Chicago.
* The relation Journey records the price of an adult fare from Chicago to the given destination by a specified operator; several operators may operate over the same route.

ANS:

1.Find the names of all destinations within 20 miles.

**𝚷destName(𝞼distance<20(Destination))**

2.List the names of all operators with at least one journey priced at under $5.

**𝚷destName( Operator ⟕ (𝞼price<5$(Journey))**

3. List the names of all operators and prices of journeys to ‘Boston’. **𝚷opeName,price(𝞼desName=Boston(Destination ⟕ Journey)⟕ Operator)**

**5.[1.5] Describe in English the relations that would be produced by the following relational algebra operations.**

1. σHotel.hotelNo = Room.hotelNo(Hotel × Room)

**List all Hotel Rooms with their Hotel information**

1. ΠhotelName(Hotel ⋈ Hotel.hotelNo = Room.hotelNo (σprice > 50 (Room)))

**List all hotel names which has rooms with price greater than $50.**

1. Guest ⋊ (σdateTo ≥ ‘1-Jan-2016’(Booking))

**Select all Guests who booked before or at jan first of 2016.**