****

**COLLEGE OF COMPUTING AND INFORMATICS**

**Assignment – 3**

**Course Title : Introduction to Database Course Code : IT244**

**Note :**

* **Submission Deadline: 16/04/2016 at 23:59**
* **5 Marks**
* **Copied assignment will be graded zero mark.**
* **Assignment submitted after due date will not be accepted; it will be considered to be late and will have zero mark.**

Q1 (a). What do you understand by normalization? (0.5 Point)

**is the process of organizing the data in database and solve the problems in database like repeat data ,data dependency ( in normalization all related stored in only one place and all linked data stored together ) normalization it is more important because allows database take small disk space and then increased performance and to improve storage efficiency, data integrity, and scalability.**

**In normalization four normal forms: first, second, third, and Boyce-Codd normal forms 1NF, 2NF, 3NF, and BCNF,*****Normalization* is a process that “improves” a database design by generating relations that are of higher normal forms.**

Q1 (b).When is a table in BCNF? (0.5 Point)

Achieved if Third normal form in addition to the following condition: It requires the absence of non key attribute defines the part of the primary key in other words It requires the lack of primary key relies on non key attribute , The table is in BCNF If found only one attribute is PK

Q1 (c). Convert following un-normalized table into normalized tables. (1 Point)

****

1NF: Remove Repeating Attributes

|  |  |  |  |
| --- | --- | --- | --- |
| PROJECT CODE | PROJECT TITL | PROJECT MANAGER | PROJECT BUDJET |
| PC010 | Pensions System | M Philips | 24500 |
| PC045 | Salaries System | H Martin | 17400 |
| PC064 | HR System | K Lewis | 12250 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Code | Employee NO. | Employee Name | Department No. | Department Name | Hourly Rate |
| PC010 | S10001 | A Smith | L004 | IT | 22:00 |
| PC010 | S10030 | L Jones | L023 | Pensions | 18:50 |
| PC010 | S21010 | P Lewis | L004 | IT | 21:00 |
| PC045 | S10010 | B Jones | L004 | IT | 21:75 |
| PC045 | S10001 | A Smith | L004 | IT | 18:00 |
| PC045 | S31002 | T Gilbert | L028 | Database | 25:50 |
| PC045 | S13210 | W Richards | L008 | Salary | 17:00 |
| PC064 | S31002 | T Gilbert | L028 | Database | 23:25 |
| PC064 | S21010 | P Lewis | L004 | IT | 17:50 |
| PC064 | S10034 | B James | L009 | HR | 16:50 |

2NF : Remove Partial Dependency

|  |  |  |  |
| --- | --- | --- | --- |
| PROJECT CODE | PROJECT TITL | PROJECT MANAGER | PROJECT BUDJET |
| PC010 | Pensions System | M Philips | 24500 |
| PC045 | Salaries System | H Martin | 17400 |
| PC064 | HR System | K lewis | 12250 |

|  |  |  |
| --- | --- | --- |
| Project Code | Employee NO. | Hourly Rate |
| PC010 | S10001 | 22:00 |
| PC010 | S10030 | 18:50 |
| PC010 | S21010 | 21:00 |
| PC045 | S10010 | 21:75 |
| PC045 | S10001 | 18:00 |
| PC045 | S31002 | 25:50 |
| PC045 | S13210 | 17:00 |
| PC064 | S31002 | 23:25 |
| PC064 | S21010 | 17:50 |
| PC064 | S10034 | 16:50 |

|  |  |  |  |
| --- | --- | --- | --- |
| Employee NO | Employee Name | Department No | Department Name |
| S10001 | A Smith | L004 | IT |
| S10030 | L Jones | L023 | Pensions |
| S21010 | P Lewis | L004 | IT |
| S10010 | B Jones | L004 | IT |
| S31002 | T Gilbert | L028 | Database |
| S13210 | W Richards | L008 | Salary |
| S10034 | B James | L009 | HR |

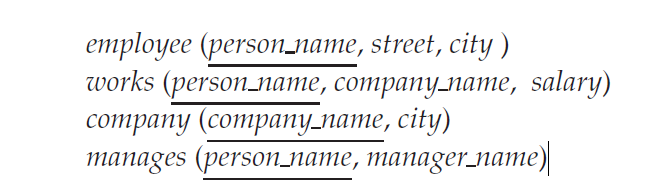
3NF: Remove transitive dependency

|  |  |  |  |
| --- | --- | --- | --- |
| PROJECT CODE | PROJECT TITL | PROJECT MANAGER | PROJECT BUDJET |
| PC010 | Pensions System | M Philips | 24500 |
| PC045 | Salaries System | H Martin | 17400 |
| PC064 | HR System | K lewis | 12250 |

|  |  |  |
| --- | --- | --- |
| Project Code | Employee NO. | Hourly Rate |
| PC010 | S10001 | 22:00 |
| PC010 | S10030 | 18:50 |
| PC010 | S21010 | 21:00 |
| PC045 | S10010 | 21:75 |
| PC045 | S10001 | 18:00 |
| PC045 | S31002 | 25:50 |
| PC045 | S13210 | 17:00 |
| PC064 | S31002 | 23:25 |
| PC064 | S21010 | 17:50 |
| PC064 | S10034 | 16:50 |

|  |  |  |
| --- | --- | --- |
| Employee NO | Employee Name | Department No |
| S10001 | A Smith | L004 |
| S10030 | L Jones | L023 |
| S21010 | P Lewis | L004 |
| S10010 | B Jones | L004 |
| S31002 | T Gilbert | L028 |
| S13210 | W Richards | L008 |
| S10034 | B James | L009 |

|  |  |
| --- | --- |
| Department No | Department Name |
| L004 | IT |
| L023 | Pensions |
| L028 | Database |
| L008 | Salary |
| L009 | HR |



Q2. Consider the relational database of the above Figure, where the primary keys are underlined. Give an expression in tuple relational calculus for each of the following queries: (1.5 Point)

1. Find all employees who work directly for “STC”

{t | ∃ r ∈ works (s[person name] = r[person name] ∧ r[company name] =" STC")}

1. Find all cities of residence of all employees who work directly for “STC”

{t | ∃ r ∈ works ∃s ∈ employee(s[person name] = r[person name] ∧ r[company name] ='STC' ∧ s[city] = t[city])}

1. Find the name of the manager of the manager of “STC”

{t | ∃ m1,m2 ∈ manages (m1[person name] = m2[manger name] ∧ m1[person name] =r[person name ] ∧ r[company name] =" STC" ∧s[manager name] = m2[manager name])}

Q3: Write the following queries in SQL, using the university schema. (1.5 Point)

classroom(building, room number, capacity)

department(dept name, building, budget)

course(course id, title, dept name, credits)

instructor(ID, name, dept name, salary)

section(course id, sec id, semester, year, building, room number, time slot id)

teaches(ID, course id, sec id, semester, year)

student(ID, name, dept name, tot cred)

takes(ID, course id, sec id, semester, year, grade)

advisor(s ID, i ID)

time slot(time slot id, day, start time, end time)

prereq(course id, prereq id)

* 1. Find the titles of courses in the Comp. Sci. department that have 3 credits.

**Select** title

**from** course

**where** dept name = ’Comp. Sci.’ and credits = 3;

* 1. Find the names of all students who have taken at least one Comp. Sci. course; make sure there are no duplicate names in the result

**select**  name

**from** student natural join takes natural join course

**where** course .dept = Comp.Sci;

* 1. Find the maximum enrollment, across all sections, in Autumn 2009.

**select** max(enrollment)

**from** (select count(ID) as enrollment

from section natural join takes

where semester = ’Autumn’ and year = 2009

group by course id, sec id);

* 1. Find the IDs and names of all students who have not taken any course offering before spring 2009

**Select** ID,names

**From**  student

**Exept**

**Selec**t ID, name

**From** student natural join takes

**where** year < 2009;

* 1. Increase the salary of each instructor in the Comp. Sci. department by 10%.

update instructor

set salary = salary \* 1.10

where dept name = ’Comp. Sci;

* 1. Delete all courses that have never been offered (that is, do not occur in the section relation).

delete from course

where course\_ id not in

(select course id from section);

* 1. Insert every student whose tot cred attribute is greater than 100 as an instructor in the same department, with a salary of $10,000.

insert into instructor

select ID, name, dept name, 10000

from student where tot cred > 100;

* 1. Create a new course “CS-001”, titled “Weekly Seminar”,with 0 credits.

Insert into course(course\_id, title, dept\_name, credits) VALUES ('CS-001',

'WeeklySeminar', 'Comp. Sci.', 0);

insert into course(course\_id, title, dept\_name, credits) values('CS-001', 'WeeklySeminar', 'Comp. Sci.', 1);

* 1. Create a section of this course in autumn 2009, with sec id of 1

Insert into section(course\_id, sec\_id, semester, year)

values ('CS-001', '1','Fall', 2009);.

* 1. Enroll every student in the Comp. Sci. department in the above section.

Insert into takes(ID, course\_id, sec\_id, semester, YEAR)

Select ID 'CS-001', '1', 'Fall', 2009

From student

where dept\_name = 'Comp. Sci.';