

1. Blank

Insert Data

-- Creating the `employee` table

CREATE TABLE employee (

employee\_name VARCHAR(50),

street VARCHAR(50),

city VARCHAR(50)

);

-- Creating the `works` table

CREATE TABLE works (

employee\_name VARCHAR(50),

company\_name VARCHAR(50),

salary INT

);

-- Inserting the data

INSERT INTO employee (employee\_name, street, city) VALUES

('Arif', '51 upashahar', 'Rajshahi'),

('Sumon', '52 east', 'Moynamati'),

('Sagor', 'Neemgachhi', 'Sirajgong'),

('Abdul', 'Binodpur', 'Rajshahi'),

('Himesh', 'Nazrul avenue', 'Dhaka'),

('Amirul', 'Chawk bazar', 'Sylhet'),

('Sajib', '99 north', 'Chittagong');

INSERT INTO works (employee\_name, company\_name, salary) VALUES

('Sumon', 'Agrani', 12000),

('Abdul', 'Sonali', 13000),

('Himesh', 'Agrani', 6000),

('Amirul', 'Sonali', 20000),

('Sagor', 'Sonali', 8000),

('Arif', 'Janata', 13000),

('Sajib', 'Janata', 9000);

1. **Find the names of all employees who live in Rajshahi city**

SELECT employee\_name

FROM employee

WHERE city = 'Rajshahi';

Ans: Abdul ,Arif

1. **Find the names and street addresses of all employees who live in Rajshahi city**

SELECT employee\_name, street

FROM employee

WHERE city = 'Rajshahi';

|  |  |
| --- | --- |
| Abdul | Binodpur |
| Arif | 51 upashahar |

1. **Find the names of all employees who work for Sonali, Agrani, and Janata**

For ALL:

SELECT employee\_name

FROM works

WHERE company\_name IN ('Sonali', 'Agrani', 'Janata');

(i)SELECT employee\_name

FROM works

WHERE company\_name = ‘Sonali’;

(ii)SELECT employee\_name

FROM works

WHERE company\_name = ‘Agrani;

(iii)SELECT employee\_name

FROM works

WHERE company\_name = ‘Janata’;

1. **Find the names and salary of all employees who work for Sonali, Agrani, and Janata**

SELECT employee\_name, salary

FROM works

WHERE company\_name IN ('Sonali', 'Agrani', 'Janata');

(i)SELECT employee\_name,salary

FROM works

WHERE company\_name = ‘Sonali’;

(ii)SELECT employee\_name,salary

FROM works

WHERE company\_name = ‘Agrani;

(iii)SELECT employee\_name,salary

FROM works

WHERE company\_name = ‘Janata’;

1. **Find the names of all employees whose salary is (i) 12000 (ii) >=12000 (iii) <12000**

-- i. Salary is 12000

SELECT employee\_name

FROM works

WHERE salary = 12000;

-- ii. Salary is >= 12000

SELECT employee\_name

FROM works

WHERE salary >= 12000;

-- iii. Salary is < 12000

SELECT employee\_name

FROM works

WHERE salary < 12000;

1. **Find the names and company of all employees whose salary is (i) 12000 (ii) >=12000 (iii) <12000**

sql

Copy code

-- i. Salary is 12000

SELECT employee\_name, company\_name

FROM works

WHERE salary = 12000;

-- ii. Salary is >= 12000

SELECT employee\_name, company\_name

FROM works

WHERE salary >= 12000;

-- iii. Salary is < 12000

SELECT employee\_name, company\_name

FROM works

WHERE salary < 12000;

1. **Find the names, streets, and cities of all employees who work for Agrani**

SELECT e.employee\_name, e.street, e.city

FROM employee e

JOIN works w ON e.employee\_name = w.employee\_name

WHERE w.company\_name = 'Agrani';

| [**employee\_name**](http://localhost/phpmyadmin/index.php?route=/sql&db=example_2&table=employee&sql_query=SELECT+e.employee_name%2C+e.street%2C+e.city%0D%0AFROM+employee+e%0D%0AJOIN+works+w+ON+e.employee_name+%3D+w.employee_name%0D%0AWHERE+w.company_name+%3D+%27Agrani%27++%0AORDER+BY+%60e%60.%60employee_name%60+ASC&sql_signature=b37deb86ff7f8e6cab787f66d5df218f3d4553c800669f1e1854d3cbdbb7f6e6&session_max_rows=25&is_browse_distinct=0) | [**street**](http://localhost/phpmyadmin/index.php?route=/sql&db=example_2&table=employee&sql_query=SELECT+e.employee_name%2C+e.street%2C+e.city%0D%0AFROM+employee+e%0D%0AJOIN+works+w+ON+e.employee_name+%3D+w.employee_name%0D%0AWHERE+w.company_name+%3D+%27Agrani%27++%0AORDER+BY+%60e%60.%60street%60+ASC&sql_signature=c63d8d58b57f9696c59a8351f0ed4344657b6f4a319fd74950766b106b3d8333&session_max_rows=25&is_browse_distinct=0) | [**city**](http://localhost/phpmyadmin/index.php?route=/sql&db=example_2&table=employee&sql_query=SELECT+e.employee_name%2C+e.street%2C+e.city%0D%0AFROM+employee+e%0D%0AJOIN+works+w+ON+e.employee_name+%3D+w.employee_name%0D%0AWHERE+w.company_name+%3D+%27Agrani%27++%0AORDER+BY+%60e%60.%60city%60+ASC&sql_signature=fd09b6c4275785b0bf26f381b2bb5ee29de8216d591d2400e9ea45690f6f273f&session_max_rows=25&is_browse_distinct=0) |  |
| --- | --- | --- | --- |
| Himesh | Nazrul avenue | Dhaka |  |
| Sumon | 52 east | Moynamati |  |

1. **Find the names, streets, and cities of all employees who earn >=10000**

SELECT e.employee\_name, e.street, e.city

FROM employee e

JOIN works w ON e.employee\_name = w.employee\_name

WHERE w.salary >= 10000;

1. **Find the names, company, and salary of all employees who live in Rajshahi city**

SELECT w.employee\_name, w.company\_name, w.salary

FROM works w

JOIN employe e ON e.employee\_name = w.employee\_name

WHERE e.city = 'Rajshahi';

| [**employee\_name**](http://localhost/phpmyadmin/index.php?route=/sql&db=example_2&table=works&sql_query=SELECT+w.employee_name%2C+w.company_name%2C+w.salary%0D%0AFROM+works+w%0D%0AJOIN+employee+e+ON+e.employee_name+%3D+w.employee_name%0D%0AWHERE+e.city+%3D+%27Rajshahi%27++%0AORDER+BY+%60w%60.%60employee_name%60+ASC&sql_signature=d369ad0f873bbfa77f44db8316e72189e2ba16743101f88a15b797570b2511b9&session_max_rows=25&is_browse_distinct=0) | [**company\_name**](http://localhost/phpmyadmin/index.php?route=/sql&db=example_2&table=works&sql_query=SELECT+w.employee_name%2C+w.company_name%2C+w.salary%0D%0AFROM+works+w%0D%0AJOIN+employee+e+ON+e.employee_name+%3D+w.employee_name%0D%0AWHERE+e.city+%3D+%27Rajshahi%27++%0AORDER+BY+%60w%60.%60company_name%60+ASC&sql_signature=82bd77b264dbcd487ba8a0ac71404ae1f8b8a8d12602a4aabfca0c0d19d37961&session_max_rows=25&is_browse_distinct=0) | [**salary**](http://localhost/phpmyadmin/index.php?route=/sql&db=example_2&table=works&sql_query=SELECT+w.employee_name%2C+w.company_name%2C+w.salary%0D%0AFROM+works+w%0D%0AJOIN+employee+e+ON+e.employee_name+%3D+w.employee_name%0D%0AWHERE+e.city+%3D+%27Rajshahi%27++%0AORDER+BY+%60w%60.%60salary%60+ASC&sql_signature=05a4d9ca8087e8091ed6f1ee4ec18c60a7a9a7fd36a511acbe86f24964bf64f1&session_max_rows=25&is_browse_distinct=0) |  |
| --- | --- | --- | --- |
| Abdul | Sonali | 13000 |  |
| Arif | Janata | 13000 |  |

1. **Find the names, streets, cities, and companies of all employees who earn >=10000**

SELECT e.employee\_name, e.street, e.city, w.company\_name

FROM employee e

JOIN works w ON e.employee\_name = w.employee\_name

WHERE w.salary >= 10000;

1. **Find the names, streets, and cities of all employees who work for Sonali and earn more than 12000**

SELECT e.employee\_name, e.street, e.city

FROM employee e

JOIN works w ON e.employee\_name = w.employee\_name

WHERE w.company\_name = 'Sonali' **AND** w.salary > 12000;

| [**employee\_name**](http://localhost/phpmyadmin/index.php?route=/sql&db=example_2&table=employee&sql_query=SELECT+e.employee_name%2C+e.street%2C+e.city%0D%0AFROM+employee+e%0D%0AJOIN+works+w+ON+e.employee_name+%3D+w.employee_name%0D%0AWHERE+w.company_name+%3D+%27Sonali%27+AND+w.salary+%3E+12000++%0AORDER+BY+%60e%60.%60employee_name%60+ASC&sql_signature=60620c1525ddce3839c4fb8bf24f53407020bbf48f925ed9bb30ac5cc7f613a4&session_max_rows=25&is_browse_distinct=0) | [**street**](http://localhost/phpmyadmin/index.php?route=/sql&db=example_2&table=employee&sql_query=SELECT+e.employee_name%2C+e.street%2C+e.city%0D%0AFROM+employee+e%0D%0AJOIN+works+w+ON+e.employee_name+%3D+w.employee_name%0D%0AWHERE+w.company_name+%3D+%27Sonali%27+AND+w.salary+%3E+12000++%0AORDER+BY+%60e%60.%60street%60+ASC&sql_signature=5f426cf4b36587b4080bd9a770d91807c6e4da22070b46df1c7ee8df48bfedcc&session_max_rows=25&is_browse_distinct=0) | [**city**](http://localhost/phpmyadmin/index.php?route=/sql&db=example_2&table=employee&sql_query=SELECT+e.employee_name%2C+e.street%2C+e.city%0D%0AFROM+employee+e%0D%0AJOIN+works+w+ON+e.employee_name+%3D+w.employee_name%0D%0AWHERE+w.company_name+%3D+%27Sonali%27+AND+w.salary+%3E+12000++%0AORDER+BY+%60e%60.%60city%60+ASC&sql_signature=b9681452b8bd92dc865791fddca49d764dedcf0962f0fdb528419fedb09266a7&session_max_rows=25&is_browse_distinct=0) |  |
| --- | --- | --- | --- |
| Abdul | Binodpur | Rajshahi |  |
| Amirul | Chawk bazar | Sylhet |  |

1. **Find all employees in the database who do not work for Sonali Bank**

SELECT e.employee\_name

FROM employee e

JOIN works w ON e.employee\_name = w.employee\_name

WHERE w.company\_name != 'Sonali';

Alternative:

SELECT employee\_name

FROM works

WHERE company\_name !='Sonali';

1. **Modify the database so that “Arif” now lives in Natore**

UPDATE employee

SET city = 'Natore'

WHERE employee\_name = 'Arif';

1. **Give all employees of “Agrani” Bank a 10 percent salary raise**

UPDATE works

SET salary = salary \* 1.10

WHERE company\_name = 'Agrani';

1. **Delete all records for Sagor in the employee table**

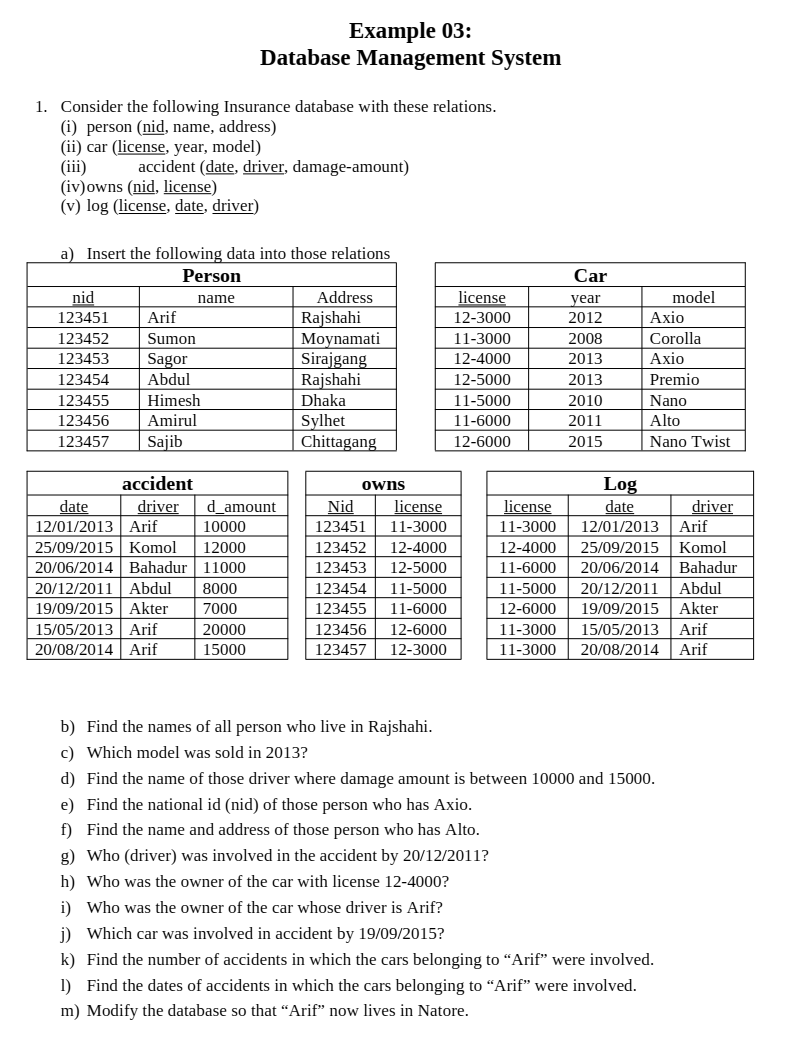
DELETE FROM employee

WHERE employee\_name = 'Sagor';

1. **Add a column manager in the works table**

ALTER TABLE works

ADD COLUMN manager VARCHAR(50);



**Example 03: Insurance Database**

**a) Insert the data:** (Similar to Example 02, use appropriate INSERT INTO queries.)

**b) Find the names of all people who live in Rajshahi:**

SELECT name FROM person WHERE address = 'Rajshahi';

Result: Arif, Abdul

**c) Which model was sold in 2013:**

SELECT model FROM car WHERE year = 2013;

Result: Axio, Premio

**d) Find the name of those drivers where the damage amount is between 10000 and 15000:**

SELECT driver

FROM accident

WHERE damage\_amount **BETWEEN 10000 AND 15000;**

Result: Arif, Bahadur

**e) Find the national ID (nid) of those who own Axio:**

SELECT nid

FROM owns

JOIN car ON **owns.license = car.license**

WHERE car.model = 'Axio';

Result: 123451, 123453

f) Find the name and address of those who own Alto:

SELECT person.name, person.address

FROM person

JOIN owns ON person.nid = owns.nid

JOIN car ON owns.license = car.license

WHERE car.model = 'Alto';

Result: Amirul, Sylhet

**g) Who was the driver involved in the accident on 20/12/2011:**

SELECT driver

FROM accident

WHERE date = '20/12/2011';

Result: Abdul

**h) Who was the owner of the car with license 12-4000:**

SELECT name

FROM person

JOIN owns ON person.nid=owns.nid

WHERE owns.license = '12-4000';

Result: Sagor

**i) Who was the owner of the car whose driver is Arif:**

SELECT name

FROM person

JOIN log ON owns.license = log.license

JOIN owns ON owns.nid=person.nid

WHERE log.driver = 'Arif';

Result: 123451, 123456, 123457

**j) Which car was involved in the accident on 19/09/2015:**

SELECT model

FROM car

JOIN log ON car.license=log.license

WHERE log.date = '19/09/2015';

Result: 11-6000

**k) Find the number of accidents in which cars belonging to Arif were involved:**

SELECT COUNT(\*)

FROM accident

WHERE driver=’Arif’

Result: 3

**l) Find the dates of accidents in which cars belonging to Arif were involved:**

SELECT date

FROM accident

Where driver=’Arif’;

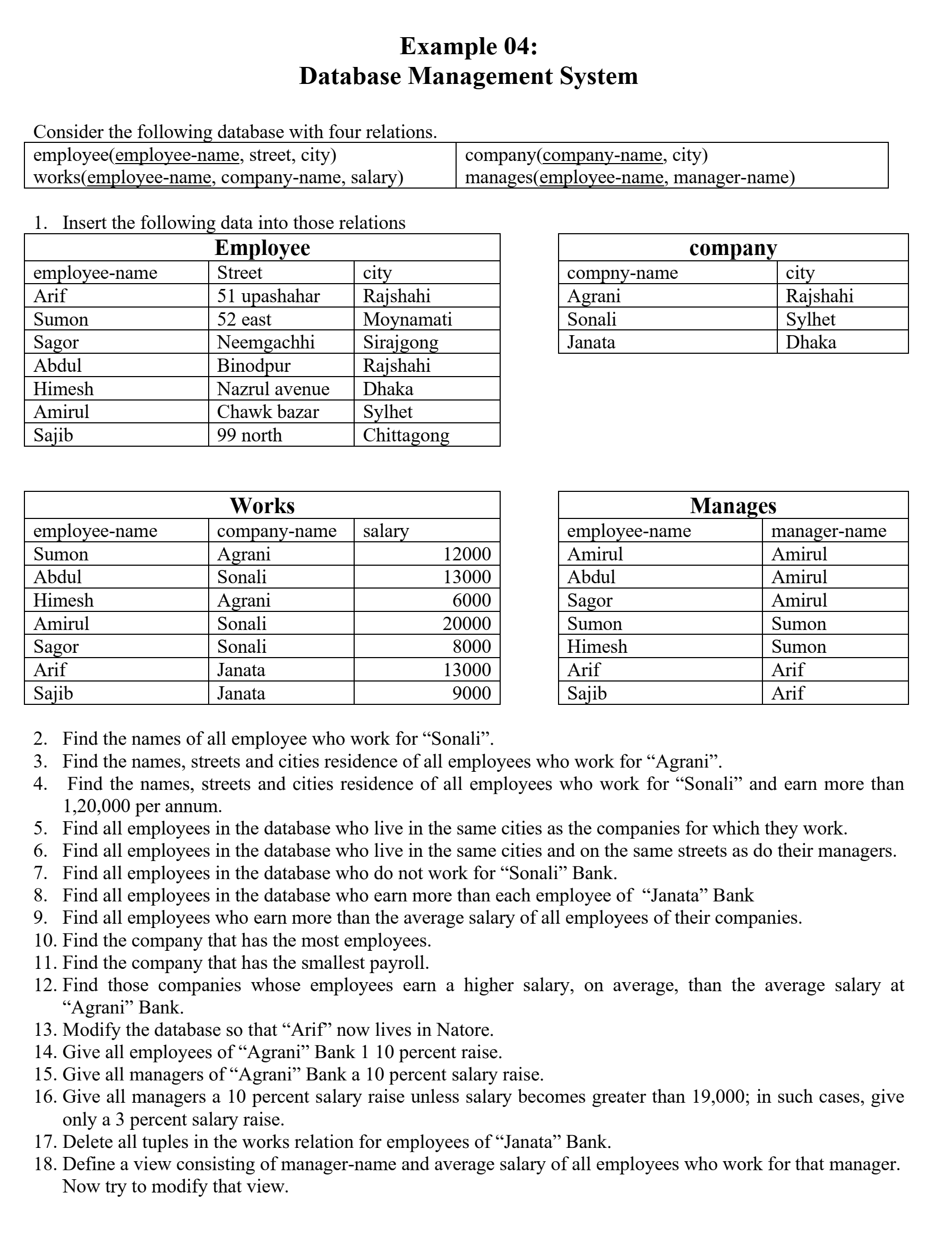
Result: 12/01/2013, 15/05/2013, 20/08/2014

**m) Modify the database so that “Arif” now lives in Natore:**

UPDATE person

SET address = 'Natore'

WHERE name = 'Arif';



**Example 04: Comprehensive Employee and Company Database**

**Schema Recap:**

1. **employee(employee-name, street, city)**
2. **works(employee-name, company-name, salary)**
3. **company(company-name, city)**
4. **manages(employee-name, manager-name)**

**a) Insert the data:**

-- Employee Table

INSERT INTO employee (employee\_name, street, city)

VALUES

('Arif', '51 upashahar', 'Rajshahi'),

('Sumon', '52 east', 'Moynamati'),

('Sagor', 'Neemgachhi', 'Sirajgong'),

('Abdul', 'Binodpur', 'Rajshahi'),

('Himesh', 'Nazrul avenue', 'Dhaka'),

('Amirul', 'Chawk bazar', 'Sylhet'),

('Sajib', '99 north', 'Chittagong');

-- Works Table

INSERT INTO works (employee\_name, company\_name, salary)

VALUES

('Sumon', 'Agrani', 12000),

('Abdul', 'Sonali', 13000),

('Himesh', 'Agrani', 6000),

('Amirul', 'Sonali', 20000),

('Sagor', 'Sonali', 8000),

('Arif', 'Janata', 13000),

('Sajib', 'Janata', 9000);

-- Manages Table

INSERT INTO manages (employee\_name, manager\_name)

VALUES

('Amirul', 'Amirul'),

('Abdul', 'Amirul'),

('Sagor', 'Amirul'),

('Sumon', 'Sumon'),

('Himesh', 'Sumon'),

('Arif', 'Arif'),

('Sajib', 'Arif');

--Company Table

INSERT INTO company (company\_name,city)

VALUES

('Agrani', 'Rajshahi'),

('Sonali', 'Sylhet'),

('Janata','Dhaka');

**Queries and Modifications**

**CREATE TABLE works**

**( employee\_name Varchar(200) ,**

**street Varchar(200),**

**city Varchar(200),**

**FOREIGN KEY employee\_name REFERENCES employee(employee\_name)**

**)**

**2. Find the names of all employees who work for Sonali Bank:**

SELECT employee\_name

FROM works

WHERE company\_name = 'Sonali';

**Result:** Abdul, Amirul, Sagor

**3. Find the names, streets, and cities of all employees who work for Agrani Bank:**

SELECT employee.employee\_name, employee.street, employee.city

FROM employee

**NATURAL JOIN works**

WHERE works.company\_name = 'Agrani';

Alternative:

SELECT employee.employee\_name, employee.street, employee.city

FROM employee

JOIN works ON employee.employee\_name = works.employee\_name

WHERE works.company\_name = 'Agrani';

Alternative:

SELECT e.employee\_name,e.street,e.city

FROM employee **AS e**

JOIN works **AS w** ON e.employee\_name = w.employee\_name

WHERE company\_name='Agrani';

**Result:**  
Sumon (52 east, Moynamati)  
Himesh (Nazrul avenue, Dhaka)

**4. Find the names, streets, and cities of all employees who work for Sonali Bank and earn more than 120000 per annum (i.e., >10000 per month):**

SELECT employee.employee\_name, employee.street, employee.city

FROM employee

JOIN works ON employee.employee\_name = works.employee\_name

WHERE works.company\_name = 'Sonali' AND works.salary > 10000;

Alternate: SELECT e.employee\_name , e.street , e.city

FROM employee AS e

JOIN works AS w

ON w.employee\_name=e.employee\_name

WHERE company\_name='Sonali' and salary>= 10000]

**Result:**  
Abdul (Binodpur, Rajshahi)  
Amirul (Chawk bazar, Sylhet)

**5. Find all employees who live in the same cities as the companies for which they work: ()**

SELECT employee.employee\_name

FROM employee

JOIN works ON employee.employee\_name = works.employee\_name

JOIN company ON works.company\_name = company.company\_name

WHERE employee.city = company.city;

Alternative:

SELECT employee.employee\_name

FROM employee

NATURAL JOIN works

NATURAL JOIN company

WHERE employee.city=company.city;

**Result:** Arif, Abdul

**6. Find all employees who live in the same cities and on the same streets as do their managers:**

SELECT e.employee\_name

FROM employee e

JOIN manages m ON e.employee\_name=m.employee\_name

JOIN employee e2 ON e2.employee\_name=m.manager\_name

WHERE e.street=e2.street AND e.city=e2.city

| [**employee\_name**](http://localhost/phpmyadmin/index.php?route=/sql&db=example_4&table=employee&sql_query=SELECT+%2A%0D%0AFROM+employee+e%0D%0AJOIN+manages+m+ON+e.employee_name%3Dm.employee_name%0D%0AJOIN+employee+e2+ON+e2.employee_name%3Dm.manager_name%0D%0AWHERE+e.street%3De2.street+AND+e.city%3De2.city++%0AORDER+BY+%60e%60.%60employee_name%60+ASC&sql_signature=b0daee38f16d09c0b3408d42973a042d08d9087f490a79dee60d0ec9744a64d6&session_max_rows=25&is_browse_distinct=0) | [**street**](http://localhost/phpmyadmin/index.php?route=/sql&db=example_4&table=employee&sql_query=SELECT+%2A%0D%0AFROM+employee+e%0D%0AJOIN+manages+m+ON+e.employee_name%3Dm.employee_name%0D%0AJOIN+employee+e2+ON+e2.employee_name%3Dm.manager_name%0D%0AWHERE+e.street%3De2.street+AND+e.city%3De2.city++%0AORDER+BY+%60e%60.%60street%60+ASC&sql_signature=4821810542f623beccc780391821a740b09392622cb8905e9c6903e38d9e7a23&session_max_rows=25&is_browse_distinct=0) | [**city**](http://localhost/phpmyadmin/index.php?route=/sql&db=example_4&table=employee&sql_query=SELECT+%2A%0D%0AFROM+employee+e%0D%0AJOIN+manages+m+ON+e.employee_name%3Dm.employee_name%0D%0AJOIN+employee+e2+ON+e2.employee_name%3Dm.manager_name%0D%0AWHERE+e.street%3De2.street+AND+e.city%3De2.city++%0AORDER+BY+%60e%60.%60city%60+ASC&sql_signature=d7d11a03c13f99b003b31cc8b722b3d2106be6bb9d2c01316d9e0529d17c43de&session_max_rows=25&is_browse_distinct=0) | [**employee\_name**](http://localhost/phpmyadmin/index.php?route=/sql&db=example_4&table=employee&sql_query=SELECT+%2A%0D%0AFROM+employee+e%0D%0AJOIN+manages+m+ON+e.employee_name%3Dm.employee_name%0D%0AJOIN+employee+e2+ON+e2.employee_name%3Dm.manager_name%0D%0AWHERE+e.street%3De2.street+AND+e.city%3De2.city++%0AORDER+BY+%60m%60.%60employee_name%60+ASC&sql_signature=39522fdd2657dfb5fd658b8c5f7b4fd2105c89f530a2c84d5bd01be32ff50bf8&session_max_rows=25&is_browse_distinct=0) | [**manager\_name**](http://localhost/phpmyadmin/index.php?route=/sql&db=example_4&table=employee&sql_query=SELECT+%2A%0D%0AFROM+employee+e%0D%0AJOIN+manages+m+ON+e.employee_name%3Dm.employee_name%0D%0AJOIN+employee+e2+ON+e2.employee_name%3Dm.manager_name%0D%0AWHERE+e.street%3De2.street+AND+e.city%3De2.city++%0AORDER+BY+%60m%60.%60manager_name%60+ASC&sql_signature=eeb935e518ea90c3fea7a10f3b6de479f032b2faa28c641c2e20e2b3f667e856&session_max_rows=25&is_browse_distinct=0) | [**employee\_name**](http://localhost/phpmyadmin/index.php?route=/sql&db=example_4&table=employee&sql_query=SELECT+%2A%0D%0AFROM+employee+e%0D%0AJOIN+manages+m+ON+e.employee_name%3Dm.employee_name%0D%0AJOIN+employee+e2+ON+e2.employee_name%3Dm.manager_name%0D%0AWHERE+e.street%3De2.street+AND+e.city%3De2.city++%0AORDER+BY+%60e2%60.%60employee_name%60+ASC&sql_signature=98688ffb14fc7b1e68a03f46798a76d96fd5a6c6296b2f88962bf38548714df7&session_max_rows=25&is_browse_distinct=0) | [**street**](http://localhost/phpmyadmin/index.php?route=/sql&db=example_4&table=employee&sql_query=SELECT+%2A%0D%0AFROM+employee+e%0D%0AJOIN+manages+m+ON+e.employee_name%3Dm.employee_name%0D%0AJOIN+employee+e2+ON+e2.employee_name%3Dm.manager_name%0D%0AWHERE+e.street%3De2.street+AND+e.city%3De2.city++%0AORDER+BY+%60e2%60.%60street%60+ASC&sql_signature=da7c68e17c1a9ea404e2158a024ebc463a1bd4908098cc045205f2ec9ba5c795&session_max_rows=25&is_browse_distinct=0) | [**city**](http://localhost/phpmyadmin/index.php?route=/sql&db=example_4&table=employee&sql_query=SELECT+%2A%0D%0AFROM+employee+e%0D%0AJOIN+manages+m+ON+e.employee_name%3Dm.employee_name%0D%0AJOIN+employee+e2+ON+e2.employee_name%3Dm.manager_name%0D%0AWHERE+e.street%3De2.street+AND+e.city%3De2.city++%0AORDER+BY+%60e2%60.%60city%60+ASC&sql_signature=9c25b6b45864b701d5a9cc3f67ceea4a00d092c8027745160d72f56edd486d1d&session_max_rows=25&is_browse_distinct=0) |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Amirul | Chawk bazar | Sylhet | Amirul | Amirul | Amirul | Chawk bazar | Sylhet |  |
| Arif | 51 upashahar | Rajshahi | Arif | Arif | Arif | 51 upashahar | Rajshahi |  |
| Sumon | 52 east | Moynamati | Sumon | Sumon | Sumon | 52 east | Moynamati |  |

**7. Find all employees in the database who do not work for Sonali Bank:**

SELECT employee\_name

FROM works

WHERE company\_name != 'Sonali';

**Result:** Sumon, Himesh, Arif, Sajib

**8. Find all employees who earn more than each employee of Janata Bank:**

SELECT works.employee\_name

FROM works

WHERE works.salary>

(SELECT MAX(works.salary)

FROM works

WHERE works.company\_name='Janata')

**Result:** Amirul

***9. Find all employees who earn more than the average salary of all employees of their company:***

SELECT employee\_name

FROM works w

WHERE w.salary >

(SELECT AVG(w2.salary)

FROM works w2

WHERE w2.company\_name=w.company\_name

)

**Result:** Amirul (Sonali), Arif (Janata),Sumon

**10. Find the company that has the most employees:**

SELECT company\_name

FROM works

GROUP BY company\_name

ORDER BY COUNT(employee\_name) DESC

LIMIT 1;

**Result:** Sonali

**11. Find the company that has the smallest payroll:**

SELECT company\_name

FROM works

GROUP BY company\_name

ORDER BY SUM(salary) ASC

LIMIT 1;

**Result:** Agrani

**12. Find those companies whose employees earn a higher salary on average than the average salary at Agrani Bank:**

SELECT company\_name

FROM works

GROUP BY company\_name

**HAVING AVG(salary)** > (SELECT AVG(salary) FROM works WHERE company\_name = 'Agrani');

**Result:** Sonali, Janata

**13. Modify the database so that “Arif” now lives in Natore:**

UPDATE employee SET city = 'Natore' WHERE employee\_name = 'Arif';

**14. Give all employees of Agrani Bank a 10% salary raise:**

UPDATE works

SET salary = salary \* 1.10

WHERE company\_name = 'Agrani';

**15. Give all managers of Agrani Bank a 10% salary raise:**

UPDATE works

SET salary=salary\*1

WHERE employee\_name IN(

SELECT manager\_name

FROM (

SELECT DISTINCT manager\_name

FROM manages

NATURAL JOIN works

WHERE company\_name='Agrani')

AS temp\_table

);

**16. Give all managers a 10% salary raise unless their salary becomes greater than 19000; in such cases give only a 3% salary raise:**

UPDATE works

**SET salary = CASE**

**WHEN salary \* 1.10 > 19000 THEN salary \* 1.03**

**ELSE salary \* 1.10**

**END**

WHERE employee\_name IN

(SELECT manager\_name FROM manages);

**17. Delete all tuples in the works relation for employees of Janata Bank:**

DELETE

FROM works

WHERE company\_name = 'Janata';

**18. Define a view consisting of manager\_name and the average salary of all employees who work for that manager. Now, try to modify that view:**

-- Create the view

CREATE VIEW manager\_avg\_salary AS

SELECT manages.manager\_name, AVG(works.salary) AS avg\_salary

FROM manages

JOIN works ON manages.employee\_name = works.employee\_name

GROUP BY manages.manager\_name;

-- Attempt to modify the view (which will generally not be allowed in most systems)

UPDATE manager\_avg\_salary SET avg\_salary = 15000 WHERE manager\_name = 'Sumon';

**Example 05:**

**Database Management System**

A relational database Schema for LIBRARY database is given below.

PublisherName Address Phone

PUBLISHER

**BookId** Title PublisherName

BOOK

BOOK\_COPIES

**BookId**  **BranchId** No\_Of\_Copies

BOOK\_LOAN

**BookId** **BranchId** **CardNo** DateOut DueDate

LIBRARY\_BRANCH

**BranchId** BranchName Address

BORROWER

**CardNo** Name Address Phone

**BookId** **AuthorName**

BOOK\_AUTHORS

Perform the following Queries on the Database

1. How many copies of the book titled DBMS are owned by the library branch whose name is “CSE Seminar library”?
2. How many copies of the book titled DBMS are owned by each library branch?
3. Retrieve the names of all borrowers who do not have any books checked out?
4. For each book that is loaned out from the “CSE seminar library” branch and whose DueDate is today, retrieve the book title, the borrower’s name, and the borrower’s address.
5. For each library branch, retrieve the branch name and the total number of books loaned out from that branch.
6. Retrieve the names, addresses, and number of books checked out for all borrowers who have more than two books checked out.
7. For each book authored by “Ivan BayRoss”, retrieve the title and the no. of copies owned by the library branch whose name is “RU central library”.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Book** | | |  | **Book\_Author** | |
| **BookId** | **Title** | **PublisherName** | **BookId** | **AuthorName** |
| 100.001cn | Computer Network | PHI | 100.001cn | A S Tanenbaum |
| 100.002dsc | Database System | Tata | 100.002dsc | Silberschatz |
| 100.003ds | Digital System | PHI | 100.003ds | Ronald J Tocci |
| 100.004db | DBMS | PHI | 100.004db | Ivan Bayross |
| 100.005ora | Oracle 2000 | Galgotia | 100.005ora | Ivan Bayross |

|  |  |  |
| --- | --- | --- |
| **Publisher** | | |
| **Name** | **Address** | **Phone** |
| PHI | 20 Delhi Super Market | 01715-454678 |
| Tata | North Kolkata | 0156-2345445 |
| Galgotia | Mumbai | 0192-203490 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Book\_Copies** | | |  | **Library\_Branch** | | | | | |
| **BookId** | **BranchId** | **No\_Of\_Copies** | **BranchId** | | **BranchName** | | | **Address** |
| 100.001cn | 1001 | 2 | 1001 | | CSE Seminar Library | | | Rajshahi |
| 100.001cn | 1002 | 5 | 1002 | | RU Central Library | | | Rajshahi |
| 100.002dsc | 1001 | 3 | 1003 | | DU Central Library | | | Dhaka |
| 100.002dsc | 1002 | 4 |  | | | | | |
| 100.003ds | 1001 | 3 | **Borrower** | | | | | |
| 100.003ds | 1003 | 5 | **CardNo** | **Name** | | **Address** | **Phone** | |
| 100.004db | 1001 | 2 | 10001 | Saidur | | CSE | 01714-400567 | |
| 100.004db | 1002 | 5 | 10002 | Rafiq | | PHYSICS | 0194-300456 | |
| 100.005ora | 1001 | 2 | 10003 | Masud | | CSE | 0156-345678 | |
| 100.005ora | 1002 | 7 | 10004 | Nobir | | ICT | 01199-203456 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **BOOK\_LOAN** | | | | |
| **BookId** | **BranchId** | **CardNo** | **DateOut** | **DueDate** |
| 100.001cn | 1001 | 10001 | 15-Jan-15 | 15-Feb-15 |
| 100.001cn | 1002 | 10002 | 25-Jan-15 | 25-Feb-15 |
| 100.002dsc | 1001 | 10003 | 20-Feb-15 | 20-Mar-15 |
| 100.002dsc | 1002 | 10004 | 15-Mar-15 | 15-Apr-15 |
| 100.003ds | 1001 | 10001 | 07-Jun-15 | 07-Jul-15 |
| 100.003ds | 1003 | 10002 | 15-Oct-15 | 15-Nov-15 |
| 100.004db | 1001 | 10003 | 25-Oct-15 | 25-Nov-15 |
| 100.004db | 1002 | 10004 | 15-Nov-15 | 15-Dec-15 |
| 100.005ora | 1001 | 10003 | 22-Dec-15 | 22-Jan-16 |
| 100.005ora | 1002 | 10001 | 25-Dec-15 | 25-Jan-16 |

Example 5

**CREATE TABLE Publisher (**

**Name VARCHAR(100) PRIMARY KEY,**

**Address VARCHAR(200),**

**Phone VARCHAR(15)**

**);**

**CREATE TABLE Book (**

**BookId VARCHAR(50) PRIMARY KEY,**

**Title VARCHAR(100) NOT NULL,**

**PublisherName VARCHAR(100) NOT NULL,**

**FOREIGN KEY (PublisherName) REFERENCES Publisher(Name)**

**);**

**CREATE TABLE Book\_Author (**

**BookId VARCHAR(50),**

**AuthorName VARCHAR(100),**

**PRIMARY KEY (BookId, AuthorName),**

**FOREIGN KEY (BookId) REFERENCES Book(BookId)**

**);**

**CREATE TABLE Library\_Branch (**

**BranchId INT PRIMARY KEY,**

**BranchName VARCHAR(100),**

**Address VARCHAR(200)**

**);**

**CREATE TABLE Borrower (**

**CardNo INT PRIMARY KEY,**

**Name VARCHAR(100),**

**Address VARCHAR(200),**

**Phone VARCHAR(15)**

**);**

**CREATE TABLE Book\_Copies (**

**BookId VARCHAR(50),**

**BranchId INT,**

**No\_Of\_Copies INT,**

**PRIMARY KEY (BookId, BranchId),**

**FOREIGN KEY (BookId) REFERENCES Book(BookId),**

**FOREIGN KEY (BranchId) REFERENCES Library\_Branch(BranchId)**

**);**

**CREATE TABLE Book\_Loan (**

**BookId VARCHAR(50),**

**BranchId INT,**

**CardNo INT,**

**DateOut DATE,**

**DueDate DATE,**

**PRIMARY KEY (BookId, BranchId, CardNo),**

**FOREIGN KEY (BookId) REFERENCES Book(BookId),**

**FOREIGN KEY (BranchId) REFERENCES Library\_Branch(BranchId),**

**FOREIGN KEY (CardNo) REFERENCES Borrower(CardNo)**

**);**

**1. Book Table**

CREATE TABLE Book (

BookId VARCHAR(50) PRIMARY KEY,

Title VARCHAR(100) NOT NULL,

PublisherName VARCHAR(100) NOT NULL,

FOREIGN KEY (PublisherName) REFERENCES Publisher(Name)

);

**2. Book\_Author Table**

CREATE TABLE Book\_Author (

BookId VARCHAR(50),

AuthorName VARCHAR(100),

PRIMARY KEY (BookId, AuthorName),

FOREIGN KEY (BookId) REFERENCES Book(BookId)

);

**3. Publisher Table**

CREATE TABLE Publisher (

Name VARCHAR(100) PRIMARY KEY,

Address VARCHAR(200),

Phone VARCHAR(15)

);

**4. Book\_Copies Table**

CREATE TABLE Book\_Copies (

BookId VARCHAR(50),

BranchId INT,

No\_Of\_Copies INT,

PRIMARY KEY (BookId, BranchId),

FOREIGN KEY (BookId) REFERENCES Book(BookId),

FOREIGN KEY (BranchId) REFERENCES Library\_Branch(BranchId)

);

**5. Library\_Branch Table**

CREATE TABLE Library\_Branch (

BranchId INT PRIMARY KEY,

BranchName VARCHAR(100),

Address VARCHAR(200)

);

**6. Borrower Table**

CREATE TABLE Borrower (

CardNo INT PRIMARY KEY,

Name VARCHAR(100),

Address VARCHAR(200),

Phone VARCHAR(15)

);

**7. Book\_Loan Table**

CREATE TABLE Book\_Loan (

BookId VARCHAR(50),

BranchId INT,

CardNo INT,

DateOut DATE,

DueDate DATE,

PRIMARY KEY (BookId, BranchId, CardNo),

FOREIGN KEY (BookId) REFERENCES Book(BookId),

FOREIGN KEY (BranchId) REFERENCES Library\_Branch(BranchId),

FOREIGN KEY (CardNo) REFERENCES Borrower(CardNo)

);

**Relationships Summary:**

* **Book** references **Publisher** (via PublisherName).
* **Book\_Author** references **Book**.
* **Book\_Copies** references **Book** and **Library\_Branch**.
* **Book\_Loan** references **Book**, **Library\_Branch**, and **Borrower**.
* **1. Publisher Table**

sql

Copy code

INSERT INTO Publisher (Name, Address, Phone)

VALUES

('PHI', '20 Delhi Super Market', '01715-454678'),

('Tata', 'North Kolkata', '0156-2345445'),

('Galgotia', 'Mumbai', '0192-203490');

**2. Book Table**

sql

Copy code

INSERT INTO Book (BookId, Title, PublisherName)

VALUES

('100.001cn', 'Computer Network', 'PHI'),

('100.002dsc', 'Database System', 'Tata'),

('100.003ds', 'Digital System', 'PHI'),

('100.004db', 'DBMS', 'PHI'),

('100.005ora', 'Oracle 2000', 'Galgotia');

**3. Book\_Author Table**

sql

Copy code

INSERT INTO Book\_Author (BookId, AuthorName)

VALUES

('100.001cn', 'A S Tanenbaum'),

('100.002dsc', 'Silberschatz'),

('100.003ds', 'Ronald J Tocci'),

('100.004db', 'Ivan Bayross'),

('100.005ora', 'Ivan Bayross');

**4. Library\_Branch Table**

sql

Copy code

INSERT INTO Library\_Branch (BranchId, BranchName, Address)

VALUES

(1001, 'CSE Seminar Library', 'Rajshahi'),

(1002, 'RU Central Library', 'Rajshahi'),

(1003, 'DU Central Library', 'Dhaka');

**5. Book\_Copies Table**

sql

Copy code

INSERT INTO Book\_Copies (BookId, BranchId, No\_Of\_Copies)

VALUES

('100.001cn', 1001, 2),

('100.001cn', 1002, 5),

('100.002dsc', 1001, 3),

('100.002dsc', 1002, 4),

('100.003ds', 1001, 3),

('100.003ds', 1003, 5),

('100.004db', 1001, 2),

('100.004db', 1002, 5),

('100.005ora', 1001, 2),

('100.005ora', 1002, 7);

**6. Borrower Table**

sql

Copy code

INSERT INTO Borrower (CardNo, Name, Address, Phone)

VALUES

(10001, 'Saidur', 'CSE', '01714-400567'),

(10002, 'Rafiq', 'PHYSICS', '0194-300456'),

(10003, 'Masud', 'CSE', '0156-345678'),

(10004, 'Nobir', 'ICT', '01199-203456');

**7. Book\_Loan Table**

sql

Copy code

INSERT INTO Book\_Loan (BookId, BranchId, CardNo, DateOut, DueDate)

VALUES

('100.001cn', 1001, 10001, '2015-01-15', '2015-02-15'),

('100.001cn', 1002, 10002, '2015-01-25', '2015-02-25'),

('100.002dsc', 1001, 10003, '2015-02-20', '2015-03-20'),

('100.002dsc', 1002, 10004, '2015-03-15', '2015-04-15'),

('100.003ds', 1001, 10001, '2015-06-07', '2015-07-07'),

('100.003ds', 1003, 10002, '2015-10-15', '2015-11-15'),

('100.004db', 1001, 10003, '2015-10-25', '2015-11-25'),

('100.004db', 1002, 10004, '2015-11-15', '2015-12-15'),

('100.005ora', 1001, 10003, '2015-12-22', '2016-01-22'),

('100.005ora', 1002, 10001, '2015-12-25', '2016-01-25');

These SQL statements create the database structure and populate it with the data. Let me know if you need additional queries or modifications!

-Bottom of Form

 **How many copies of the book titled DBMS are owned by the library branch named "CSE Seminar Library"?**

Query:

SELECT SUM(bc.no\_of\_Copies)

FROM book\_copies AS bc

NATURAL JOIN book as b

NATURAL JOIN library\_branch as lb

WHERE lb.BranchName='CSE Seminar Library' AND b.Title='DBMS';

 **How many copies of the book titled DBMS are owned by each library branch?**

Query:

SELECT lb.BranchName,b.Title,bc.no\_of\_Copies

FROM book\_copies AS bc

NATURAL JOIN book as b

NATURAL JOIN library\_branch as lb

WHERE b.Title='DBMS';

Alternative:

SELECT lb.BranchName,b.Title,SUM(bc.no\_of\_Copies)

FROM book\_copies AS bc

NATURAL JOIN book as b

NATURAL JOIN library\_branch as lb

WHERE b.Title='DBMS'

GROUP BY lb.BranchName;

 **Retrieve the names of all borrowers who do not have any books checked out.**

Query:

SELECT Name

FROM Borrower

WHERE CardNo NOT IN (SELECT CardNo FROM Book\_Loan);

Alternative:

SELECT br.Name

FROM borrower as br

LEFT JOIN book\_loan as bl ON br.CardNo=bl.CardNo

WHERE bl.BookId IS NULL;

 **For each book loaned out from the “CSE Seminar Library” branch and whose DueDate is today, retrieve the book title, borrower’s name, and the borrower’s address.**

Query:

SELECT b.title, br.Name,br.Address,br.Phone

FROM Borrower as br

JOIN book\_loan as bl ON br.CardNo=bl.CardNo

JOIN library\_branch as lb ON lb.BranchId=bl.BranchId

JOIN book as b ON b.BookId=bl.BookId

WHERE lb.BranchName='CSE Seminar Library' AND bl.DueDate='2015-02-15';

 **For each library branch, retrieve the branch name and the total number of books loaned out from that branch.**

Query:

SELECT lb.branchname , sum(bc.no\_of\_copies)

FROM library\_branch as lb

NATURAL JOIN book\_copies as bc

GROUP BY lb.branchName;

 **Retrieve the names, addresses, and number of books checked out for all borrowers who have more than two books checked out.**

Query:

SELECT br.Name,br.Address,br.Phone

FROM borrower as br

NATURAL JOIN book\_loan as bl

GROUP BY bl.CardNo

HAVING COUNT(bl.CardNo)>2;

 **For each book authored by “Ivan BayRoss,” retrieve the title and the number of copies owned by the library branch named “RU Central Library.”**

Query:

SELECT b.Title , SUM(bc.no\_of\_copies)

FROM Book as b

JOIN book\_author as ba ON b.BookId=ba.BookId

JOIN book\_copies as bc ON bc.BookId=ba.BookId

JOIN library\_branch as lb ON bc.BranchId=lb.BranchId

WHERE ba.AuthorName='Ivan Bayross' AND lb.BranchName='RU Central Library'

GROUP BY bc.BookId;