Database Management System – cs422 DE

Lab 2 – Week 5

This Lab is based on lecture 5 (chapters 14).

- Submit your *own work* on time. No credit will be given if the lab is submitted after the due date.
- Note that the completed lab should be submitted in .doc, .docx, .rtf or .pdf format only.
- If you think that your answer needs more explanation to get credit then please write it down.

Consider a relation with following attributes:

EmpNo: Employee Number
EmpName : Employee Name
EmpEmail : Employee Email
ProjNo : Project Number
ProjName : Project Name
EmpGrade : Employee Grade
HrlyRate : Hourly rate of compensation
Employees of the same grade receive the same hourly compensation
HrsWorked : Hours a particular employee worked on a particular project

1. Create this table and sample data in SQL Server. There must be at least 10 rows. There must be 3 to 6 Employees and 3 to 6 projects. You need to add the screenshot of the table showing all the rows.

```
SELECT * FROM EmpProjectWork;
```

empprojectwork 1

SELECT * FROM EmpProjectWork | Enter a SQL expression to filter results (use Ctrl+Space)

	empno	empname	empemail	projno	projname	empgrade	hrlyrate	hrsworked
1	101	Alice Lee	alice@corp.com	10	Website Revamp	G1	50	16
2	101	Alice Lee	alice@corp.com	11	Mobile App	G1	50	12
3	101	Alice Lee	alice@corp.com	12	Data Warehouse	G1	50	20
4	102	Bob Tran	bob@corp.com	10	Website Revamp	G2	40	15
5	102	Bob Tran	bob@corp.com	13	API Gateway	G2	40	10
6	103	Carol Nguyen	carol@corp.com	11	Mobile App	G2	40	18
7	103	Carol Nguyen	carol@corp.com	12	Data Warehouse	G2	40	8
8	104	David Vo	david@corp.com	10	Website Revamp	G3	30	25
9	104	David Vo	david@corp.com	13	API Gateway	G3	30	5
10	104	David Vo	david@corp.com	11	Mobile App	G3	30	6

2. Find all functional dependencies.
   ANS: EmpNo -> EmpName, EmpEmail, EmpGrade; EmpGrade -> HrlyRate; ProjNo -> ProjName; (EmpNo, ProjNo) -> HrsWorked.

3. Find all Candidate Keys.
   ANS: (EmpNo, ProjNo)

4. Find a Primary Key.
   ANS: (EmpNo, ProjNo)

5. Find all partial dependencies.
   ANS: EmpNo -> EmpName, EmpEmail, EmpGrade; ProjNo -> ProjName.

6. Normalize to 2NF.
   ANS:
   Decompose the original relation (EmpNo, EmpName, EmpEmail, EmpGrade, HrlyRate, ProjNo, ProjName, HrsWorked) into:

   EMPLOYEE(EmpNo, EmpName, EmpEmail, EmpGrade, HrlyRate) – PK: EmpNo
   PROJECT(ProjNo, ProjName) – PK: ProjNo
   EMP_PROJECT(EmpNo, ProjNo, HrsWorked) – PK: (EmpNo, ProjNo), FKs: EmpNo -> EMPLOYEE, ProjNo -> PROJECT

   All non-key attributes in each table now depend on the whole key of that table (no partial dependencies).

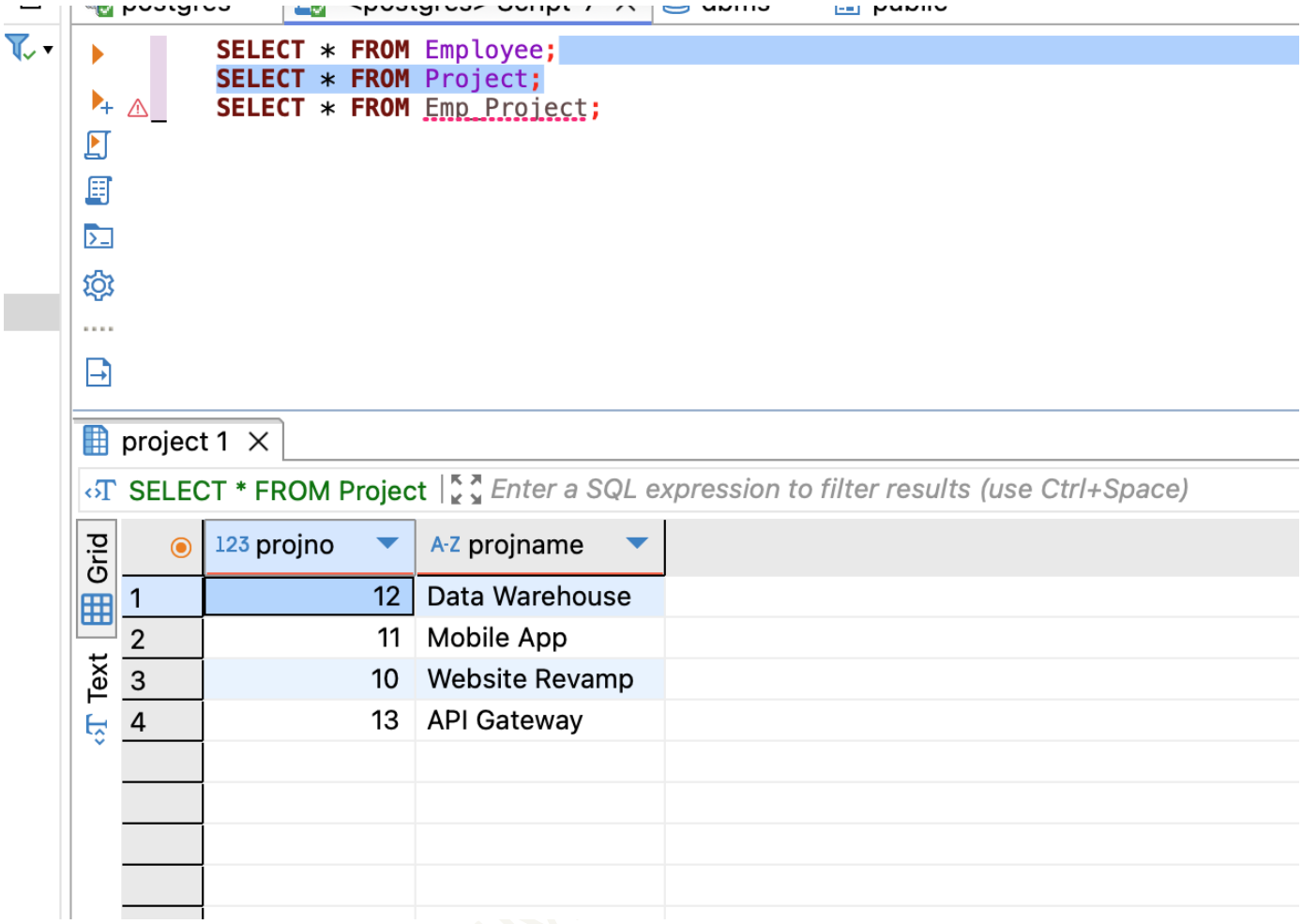
7. Show new tables after 2NF (based on the sample data you created in 1 above). Screenshots of all the tables are required.
   ANS:

```
SELECT * FROM Employee;
SELECT * FROM Project;
SELECT * FROM Emp_Project;
```

employee 1

SELECT * FROM Employee | Enter a SQL expression to filter results (use Ctrl+Space)

	123 empno	A-Z empname	A-Z empemail	A-Z empgrade	123 hrlyrate
1	104	David Vo	david@corp.com	G3	30
2	103	Carol Nguyen	carol@corp.com	G2	40
3	101	Alice Lee	alice@corp.com	G1	50
4	102	Bob Tran	bob@corp.com	G2	40


SELECT * FROM Employee;
SELECT * FROM Project;
SELECT * FROM Emp_Project;
project 1
SELECT * FROM Project
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
Text
123 projno
A-Z projname
1 12 Data Warehouse
2 11 Mobile App
3 10 Website Revamp
4 13 API Gateway

```
SELECT * FROM Project;
SELECT * FROM Emp_Project;
```

emp_project 1

SELECT * FROM Emp_Project | Enter a SQL expression to filter results (use Ctrl+Space)

Grid | Text

	123 empno	123 projno	123 hrsworked
1	101	10	16
2	101	11	12
3	101	12	20
4	102	10	15
5	102	13	10
6	103	11	18
7	103	12	8
8	104	10	25
9	104	13	5
10	104	11	6

8. Normalize to 3NF.
   ANS:

   Remove the transitive dependency EmpNo -> EmpGrade -> HrlyRate by separating grade and rate:
   EMPLOYEE(EmpNo, EmpName, EmpEmail, EmpGrade) – PK: EmpNo, FK EmpGrade -> GRADE
   GRADE(EmpGrade, HrlyRate) – PK: EmpGrade
   PROJECT(ProjNo, ProjName) – PK: ProjNo
   EMP_PROJECT(EmpNo, ProjNo, HrsWorked) – PK: (EmpNo, ProjNo), FKs: EmpNo -> EMPLOYEE, ProjNo -> PROJECT
   Each non-key attribute now depends only on the key, the whole key, and nothing but the key (3NF).

9. Show new tables after 3NF (based on the sample data you created in 1 above). Screenshots of all the tables are required.

```
SELECT * FROM Employee;
SELECT * FROM Grade;
SELECT * FROM Project;
SELECT * FROM Emp_Project;
```

employee 1

SELECT * FROM Employee | Enter a SQL expression to filter results (use Ctrl+Space)

Grid | Text

	empno	empname	empemail	empgrade
1	104	David Vo	david@corp.com	G3
2	103	Carol Nguyen	carol@corp.com	G2
3	101	Alice Lee	alice@corp.com	G1
4	102	Bob Tran	bob@corp.com	G2

4K
4K
8K
4K

```
SELECT * FROM Employee;
SELECT * FROM Grade;
SELECT * FROM Project;
SELECT * FROM Emp_Project;
```

grade 1

SELECT * FROM Grade | Enter a SQL expression to filter results (use Ctrl+Space)

Grid | Text

	empgrade	hrlyrate
1	G2	40
2	G3	30
3	G1	50

```sql
SELECT * FROM Employee;
SELECT * FROM Grade;
SELECT * FROM Project;
SELECT * FROM Emp_Project;
```

project 1

SELECT * FROM Project | Enter a SQL expression to filter results (use Ctrl+Sp

Grid | Text

	projno	projname
1	12	Data Warehouse
2	11	Mobile App
3	10	Website Revamp
4	13	API Gateway

```sql
SELECT * FROM Employee;
SELECT * FROM Grade;
SELECT * FROM Project;
SELECT * FROM Emp_Project;
```

emp_project 1

SELECT * FROM Emp_Project | Enter a SQL expression to filter results (use Ctrl+Space)

Grid | Text

	empno	projno	hrsworked
1	101	10	16
2	101	11	12
3	101	12	20
4	102	10	15
5	102	13	10
6	103	11	18
7	103	12	8
8	104	10	25
9	104	13	5
10	104	11	6