

## 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.

EmpNo	EmpName	EmpEmail	ProjNo	ProjName	EmpGrade	HrlyRate	HrsWorked
1	John Doe	john.doe@example.com	101	Project A	1	20.00	
2	Jane Smith	jane.smith@example.com	102	Project B	2	25.00	
3	Michael Johnson	michael.johnson@example.com	103	Project C	1	20.00	
4	Emily Brown	emily.brown@example.com	101	Project A	3	30.00	
5	William Wilson	william.wilson@example.com	102	Project B	2	25.00	
6	Sophia Lee	sophia.lee@example.com	103	Project C	3	30.00	
7	Daniel Taylor	daniel.taylor@example.com	101	Project A	1	20.00	
8	Olivia Martinez	olivia.martinez@example.com	102	Project B	2	25.00	
9	James Anderson	james.anderson@example.com	103	Project C	3	30.00	
10	Emma Garcia	emma.garcia@example.com	101	Project A	1	20.00	
11	Alexander Hernandez	alexander.hernandez@example.com	102	Project B	2	25.00	
12	Mia Lopez	mia.lopez@example.com	103	Project C	3	30.00	
13	Ethan Gonzalez	ethan.gonzalez@example.com	101	Project A	1	20.00	
14	Isabella Perez	isabella.perez@example.com	102	Project B	2	25.00	
15	Aiden Torres	aiden.torres@example.com	103	Project C	3	30.00	

2. Find all functional dependencies.

ANS:

- (EmpNo) -> EmpName, EmpEmail, EmpGrade, HrlyRate
- (ProjNo) -> ProjName
- (EmpGrade) -> HrlyRate
- (EmpNo, ProjNo) -> HrsWorked

3. Find all Candidate Keys.

ANS:

- (EmpNo, ProjNo) -> uniquely identify all other attributes.

4. Find a Primary Key.

ANS:

- (EmpNo, ProjNo)

5. Find all partial dependencies.

ANS:

- (EmpNo) -> EmpName, EmpEmail, EmpGrade, HrlyRate
- (ProjNo) -> ProjName

6. Normalize to 2NF.

ANS: Eliminate partial dependencies.

- Employee (EmpNo, EmpName, EmpEmail, EmpGrade, HrlyRate)
- Project (ProjNo, ProjName)
- EmployeeProject (EmpNo, ProjNo, HrsWorked)

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

	ProjNo	ProjName
1	101	Project A
2	102	Project B
3	103	Project C

Output master.dbo.Project master.dbo.Employee master.dbo.EmployeeProject				
15 rows Tx: Auto DDL				
	EmpNo	ProjNo	HrsWorked	
1	1	101	40	
2	2	102	35	
3	3	103	45	
4	4	101	50	
5	5	102	38	
6	6	103	42	
7	7	101	48	
8	8	102	41	
9	9	103	47	
10	10	101	55	
11	11	102	37	
12	12	103	43	
13	13	101	52	
14	14	102	40	
15	15	103	46	

  

Output master.dbo.Project master.dbo.Employee master.dbo.EmployeeProject					
15 rows Tx: Auto DDL					
	EmpNo	EmpName	EmpEmail	EmpGrade	HrlyRate
1	1	John Doe	john.doe@example.com	1	20.00
2	2	Jane Smith	jane.smith@example.com	2	25.00
3	3	Michael Johnson	michael.johnson@example.com	1	20.00
4	4	Emily Brown	emily.brown@example.com	3	30.00
5	5	William Wilson	william.wilson@example.com	2	25.00
6	6	Sophia Lee	sophia.lee@example.com	3	30.00
7	7	Daniel Taylor	daniel.taylor@example.com	1	20.00
8	8	Olivia Martinez	olivia.martinez@example.com	2	25.00
9	9	James Anderson	james.anderson@example.com	3	30.00
10	10	Emma Garcia	emma.garcia@example.com	1	20.00
11	11	Alexander Hernandez	alexander.hernandez@example.com	2	25.00
12	12	Mia Lopez	mia.lopez@example.com	3	30.00
13	13	Ethan Gonzalez	ethan.gonzalez@example.com	1	20.00
14	14	Isabella Perez	isabella.perez@example.com	2	25.00
15	15	Aiden Torres	aiden.torres@example.com	3	30.00

8. Normalize to 3NF.

ANS: Eliminate transitive dependency: (EmpGrade) -> HrlyRate

- Employee (EmpNo, EmpName, EmpEmail, EmpGrade)

- EmpCompensation (EmpGrade, HrlyRate)
- Project (ProjNo, ProjName)
- EmployeeProject (EmpNo, ProjNo, HrsWorked)

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

console [localhost] x Employee [localhost] x

15 rows v

WHERE ORDER BY

	EmpNo	EmpName	EmpEmail	EmpGrade
1	1	John Doe	john.doe@example.com	1
2	2	Jane Smith	jane.smith@example.com	2
3	3	Michael Johnson	michael.johnson@example.com	1
4	4	Emily Brown	emily.brown@example.com	3
5	5	William Wilson	william.wilson@example.com	2
6	6	Sophia Lee	sophia.lee@example.com	3
7	7	Daniel Taylor	daniel.taylor@example.com	1
8	8	Olivia Martinez	olivia.martinez@example.com	2
9	9	James Anderson	james.anderson@example.com	3
10	10	Emma Garcia	emma.garcia@example.com	1
11	11	Alexander Hernandez	alexander.hernandez@example.com	2
12	12	Mia Lopez	mia.lopez@example.com	3
13	13	Ethan Gonzalez	ethan.gonzalez@example.com	1
14	14	Isabella Perez	isabella.perez@example.com	2
15	15	Aiden Torres	aiden.torres@example.com	3

console [localhost] x Project [localhost] x

3 rows v

WHERE

	ProjNo	ProjName
1	101	Project A
2	102	Project B
3	103	Project C

console [localhost] × EmpCompensation [localhost] ×

3 rows

WHERE ORDER BY

	EmpGrade	HrlyRate
1	1	20.00
2	2	25.00
3	3	30.00

console [localhost] × EmployeeProject [localhost] ×

15 rows

WHERE ORDER BY

	EmpNo	ProjNo	HrsWorked
1	1	101	40
2	2	102	35
3	3	103	45
4	4	101	50
5	5	102	38
6	6	103	42
7	7	101	48
8	8	102	41
9	9	103	47
10	10	101	55
11	11	102	37
12	12	103	43
13	13	101	52
14	14	102	40
15	15	103	46