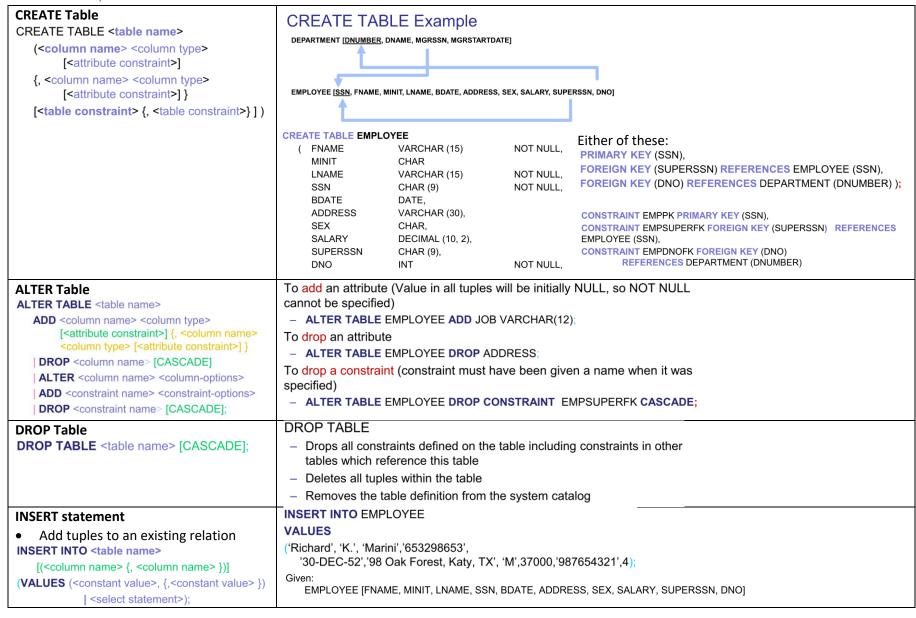
Module 7: MySQL



	INSERT INTO DEPTS-INFO (DNAME, NO-OF-EMPLOYEES, TOTAL-SALARY) SELECT DNAME, COUNT (*), SUM (SALARY) FROM DEPARTMENT, EMPLOYEE WHERE DNUMBER = DNO GROUP BY DNAME;				
 DELETE statement Remove existing tuples from a relation DELETE FROM [WHERE <select condition="">];</select> 	Given: EMPLOYEE [FNAME, MINIT, LNAME, SSN, BDATE, ADDRESS, SEX, SALARY, SUPERSSN, DNO] DEPARTMENT [DNAME, DNUMBER, MGRSSN, MGRSTARTDATE] DELETE FROM EMPLOYEE WHERE DNO = 5; Given: EMPLOYEE [FNAME, MINIT, LNAME, SSN, BDATE, ADDRESS, SEX, SALARY, SUPERSSN, DNO]				
 UPDATE statement Modify attribute values of one/more selected tuples in a relation UPDATE SET <column name=""> = <value expression=""> {, <column name=""> = <value expression="">}</value></column></value></column> [WHERE <select condition="">];</select> 	UPDATE EMPLOYEE SET SALARY = SALARY * 1.1 WHERE LNAME = 'McGowen'; Given: EMPLOYEE [FNAME, MINIT, LNAME, SSN, BDATE, ADDRESS, SEX, SALARY, SUPERSSN, DNO]				
SELECT statement SELECT <attribute list=""> FROM [WHERE <condition>];</condition></attribute>	SELECT EMP_ADDRESS FROM EMPLOYEE WHERE EMP_NAME = "Joe Bates" EMPLOYEE				
ORDER BY	EMP_NAME Nicole Smith Joe Bates	EMP_ADDRESS 1 Pine Road 32 Chandler Rd	DEPARTMENT Info. Systems Manufacturing		
	D 11	EMP_ADDRESS 32 Chandler Rd			

Complex WHERE conditions

Substring Comparisons	Examples		
LIKE	WHERE Address LIKE '%StLucia%'		
	• WHERE StrDate LIKE ' / 05/'		
IN	WHERE LName IN ('Jones', 'Wong', ' Harrison')		
IS	WHERE DNo is NULL		
(usually used in conjunction			
with NULL and NOT NULL			
Arithmetic Operators &			
Functions			
+, -, *, /, date and time	• WHERE Salary * 2 > 5000		
functions	 WHERE Year(Sys_Date – Bdate) > 55		
BETWEEN	WHERE Salary BETWEEEN 10000 AND 30000		
Other functions			
DISTINCT	9. List the distinct salaries paid to employees in each depar	tment	_
 Removes duplicates 			
	SELECT DISTINCT Salary, DNo	Salary	Dno
	FROM EMPLOYEE;	30000	5
		40000	5
		25000	4
		43000	4
		25000	5
		55000	1

Sorting Example - By Heading

14b. List the last names of all employees working in department 6, and their salaries given a 10% increase.

SELECT Lname, 1.1 * Salary AS Inc-Sal FROM EMPLOYEE
WHERE Dno = 6
ORDER BY Inc-Sal;

Given:

EMPLOYEE [FNAME, MINIT, LNAME, SSN, BDATE, ADDRESS, SEX, SALARY, SUPERSSN, DNO]

14a. List the last names of all employees working in department 6, and their salaries given a 10% increase.

SELECT Lname, 1.1 * Salary FROM EMPLOYEE WHERE Dno = 6 ORDER BY 2;

Given:

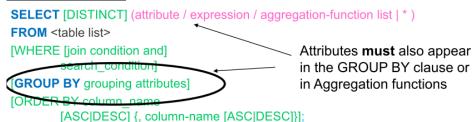
EMPLOYEE [FNAME, MINIT, LNAME, SSN, BDATE, ADDRESS, SEX, SALARY, SUPERSSN, DNO]

Aggregation

 Functions that produce summary values, which can be applied to a selected set of tuples, to all tuples, or to multiple groups of tuples, specified by the GROUP BY clause

COUNT	Counts the number of tuples query returns	
SUM	Calculates the sum of a set of numeric values	
AVG	Calculates the average of a set of numeric values	
MAX	Returns the maximum value from a set of values, which have a total ordering. (Note that domain	
	of values can be non-numeric)	
MIN	Returns the minimum value from a set of values, which have a total ordering. (Note that domain	
	of values can be non-numeric)	

GROUPING in SQL



When GROUP BY is used in an SQL statement, any attribute appeared in SELECT Clause must also appeared
in an aggregation function or in GROUP BY clause.

Conditions on Groups

HAVING clause (following the GROUP BY clause) is used to specify the conditions (similar to WHERE clause), but can also include aggregates

Set Operators

UNION	• Produces a relation that includes all tuples that appear only in R1, or only in R2, or in both and R2.	
	Duplicate entries are eliminated.	
	Two relations are union compatible if:	
	they have the same no. of columns	
	Their columns have corresponding domains (i.e. dom(Ai) = dom(Bi))	

25. List the ESSN's of employees who have dependents or work on projects

SELECT ESSN FROM WORKS ON

UNION

SELECT ESSN FROM DEPENDENTS

Given:

WORKS_ON [ESSN, PNo, Hours]
DEPENDENT [ESSN, Dep. Name, Sex, DOB, Relationship]

25a. List the ESSN's of employees who have dependents or work on projects

SELECT ESSN FROM WORKS_ON

UNION ALL

SELECT ESSN FROM DEPENDENTS

Result is a multi-set (containing duplicate tuples).
Can also be applied to Intersection and Difference
Given:

WORKS_ON [ESSN, PNo, Hours]

DEPENDENT [ESSN, Dep Name, Sex, DOB, Relationship]

Intersection

- Produces a relation that includes the tuples that appear in both R1 and R2
- R1 & R2 must be union compatible.

27. List the ESSN's of employees who have dependents and work on projects

SELECT ESSN FROM WORKS ON

INTERSECT

SELECT ESSN FROM DEPENDENTS

Given:

WORKS_ON [ESSN, PNo, Hours] DEPENDENT [ESSN, Dep_Name, Sex, DOB, Relationship]

Difference

- Produces a relation that includes all the tuples that appear in R1, but do not appear in R2.
- R1 and R2 must be union compatible.
- 28. List the ESSN's of employees who have dependents but do not work on projects

SELECT ESSN FROM DEPENDENTS

MINUS

SELECT ESSN FROM WORKS_ON

Given:

WORKS_ON [ESSN, PNo, Hours]

DEPENDENT [ESSN, Dep Name, Sex, DOB, Relationship]

A∪B	commutative	$A \cup B = B \cup A$
	associative	$(A \cup B) \cup C = A \cup (B \cup C)$
A∩B	commutative	$A \cap B = B \cap A$
	associative	$(A \cap B) \cap C = A \cap (B \cap C)$
A-B	not commutative	$A-B \neq B-A$
	not associative	$(A-B)-C \neq A-(B-C)$

EXAMPLE (commutative):

SELECT * FROM WORKS_ON UNION SELECT * FROM WORKED_ON

= SELECT * FROM WORKED_ON UNION SELECT * FROM WORKS_ON

Renaming in SQL

Qualifying attributes names or declaring an alias

29. Get employee names and the corresponding department names

SELECT EMPLOYEE.Name, DEPARTMENT.Name FROM EMPLOYEE. DEPARTMENT WHERE EMPLOYEE.Dno = DEPARTMENT.Dnumber

EMPLOYEE [NAME, SSN, BDATE, ADDRESS, SEX, SALARY, SUPERSSN, DNo] DEPARTMENT [NAME, DNUMBER, MGRSSN, MGRSTARTDATE]

Foreign Keys:

EMPLOYEE.SUPERSSN → EMPLOYEE.SSN EMPLOYEE.DNO → DEPARTMENT.DNUMBER DEPARTMENT.MGRSSN → EMPLOYEE.SSN

Declaring an Alias

30. Get employee names and the corresponding supervisor names

SELECT SUBORDINATE.Name, SUPERVISOR.Name FROM EMPLOYEE AS SUBORDINATE, EMPLOYEE AS SUPERVISOR WHERE SUBORDINATE.SuperSSN = SUPERVISOR.SSN

Attributes can also be renamed using AS

Given:

EMPLOYEE [NAME, SSN, BDATE, ADDRESS, SEX, SALARY, SUPERSSN, DNUMBER]

Foreign Key:

EMPLOYEE.SUPERSSN → EMPLOYEE.SSN