## Lecture 6 practice – SQL

Part I. Consider the following relations:

## Hotel (hotelNo, hotelName, city)

## Room (roomNo, hotelNo, type, price)

## Booking (hotelNo, guestNo, dateFrom, dateTo, roomNo)

## Guest (guestNo, guestName, guestAddress)

(The underlined attributes form the primary key. Note that SQL has a DATE domain that dateFrom and dateTo use. Dates can be compared using operators, such as < and >.)

Write SQL statements to perform the following (you may use hotel.db file on Canvas Files/practices to try):

1. Display all information in the hotel relation.

SELECT \* FROM HOTEL;

1. List the names and addresses of all guests from Seattle, alphabetically by name. (Assume the guestAddress in Guest table contains the information including city, state, etc.)

SELECT guestName, guestAddress

FROM GUEST

WHERE guestAddress LIKE '%Seattle%'

ORDER BY guestName;

1. Determine the number of hotels in the database.

SELECT COUNT(\*) FROM HOTEL;

1. Determine how many different guests have made bookings for some part of March 2015.

SELECT COUNT(DISTINCT guestNo)

FROM BOOKING

WHERE dateFrom LIKE '%March 2015%' OR dateTo LIKE '%March 2015%';

1. Count how many hotels there are in each city.

SELECT DISTINCT city, COUNT(hotelNo) AS hotel\_number

FROM HOTEL

GROUP BY city;

1. List the hotelNo and average price of ‘Standard’ rooms at each hotel.

SELECT hotelNo, AVG(price)

FROM ROOM

GROUP BY hotelNo

HAVING type='Standard';

1. List the name of all guests currently staying at the Hilton (use CURRENT\_DATE – see below).

SELECT guestName

FROM GUEST

WHERE dateFrom <= CURRENT\_DATE AND dateTo >= CURRENT\_DATE;

GROUP BY guestName

HAVING hotelName='Hilton';

1. Count the number of rooms in each hotel in Las Vegas along with the name of the hotel.

SELECT hotelName, COUNT(roomNo)

FROM HOTEL, ROOM

WHERE city='Las Vegas';

1. List hotelNo who have more than 2 ‘Double’ rooms.

SELECT hotelNo, type, COUNT(\*)

FROM ROOM

WHERE type='Double'

GROUP BY hotelNo

HAVING COUNT(\*)>2;

1. List the rooms that are currently unoccupied at the Hilton.

SELECT roomNo

FROM ROOM

WHERE NOT EXISTS (SELECT \*

FROM BOOKING

WHERE ROOM.roomNo=BOOKING.roomNo);

**Query syntax:**

SELECT [DISTINCT] {\* | [columnExpression [AS newName]] [,…]}

FROM TableName [alias] [,…]

[WHERE condition]

[GROUP BY columnList

[HAVING condition] ]

[ORDER BY columnList]

CURRENT\_DATE returns the current date in the time zone that is local to the user.

Part II



Get the total number of employees whose salaries exceed $30,000 in each department, but only for departments where more than 2 employees work

SELECT Dname, COUNT(\*)

FROM EMPLOYEE , DEPARTMENT

WHERE Dnumber=Dno AND salary > 30000 AND Dno IN

(SELECT Dno

FROM EMPLOYEE

GROUP BY Dno

HAVING COUNT(\*)>2)

GROUP BY Dname;

## Part III. Update

1. Any syntax errors? Assume test & test2 both have 3 attributes: id int, name varchar(15), salary decimal(10,2)

1. Insert table test values(1, John, 30)

Insert into test values(1, 'John', 30);

1. Insert into test values (2, 'John');

Insert into test values(2, 'John', NULL);

1. Insert into test values as select \* from test2;

Insert into test values select \* from test2;

1. Delete from table test, test2 where id = 1;

Delete from test where id = 1;

Delete from test2 where id = 1;

1. Update set test.id = 3 where name = 'John';

Update test set id = 3 where name = 'John';

2. Write SQL statement to increase the salary of John Smith by 25% (of the Company.db)

SELECT Fname, Lname, salary + (salary \* 0.25) AS increased\_salary

FROM EMPLOYEE

WHERE Fname='John' AND Lname='Smith';

**Syntax:**

**INSERT INTO** table\_name [ ( col\_name1, col\_name2, .... ) ]

**VALUES** ( expression1\_1, expression1\_2, .... ), ( expression2\_1, expression2\_2, .... ), ....

**INSERT INTO** table\_name [ ( col\_name1, col\_name2, .... ) ]

**SELECT** ...

**DELETE FROM** table\_name

**[WHERE** expression]

**UPDATE** table\_name

**SET** col\_name1 **=** expression1, col\_name2 = expression2, ....

[**WHERE** expression ]