## 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:

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;

Or ignore case:

SELECT guestName, guestAddress FROM Guest WHERE guestaddress LIKE '%seattle%' collate nocase 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 <='2015-03-01' and dateTo >= '2015-03-01') OR

(dateFrom>='2015-03-01' and dateFrom<='2015-03-31');

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

Select city, count(\*) from Hotel group by city;

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

Select hotelNo, avg(price) from Room where type = 'Standard' group by hotelNo;

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

SELECT guest.guestno, guestname, guestaddress

FROM guest, booking, hotel

WHERE guest.guestno =booking.guestno AND

hotel.hotelno = booking.hotelno AND

(datefrom <= CURRENT\_DATE

AND dateto >= CURRENT\_DATE) AND

hotelname = 'Hilton';

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

SELECT H.hotelNo, hotelName, COUNT(roomNo) FROM Room R, Hotel H

Where R.hotelNo= H.hotelNo and City like '%Las Vegas%'

GROUP BY h.hotelNo, hotelName;

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

***Select hotelNo, count(type)***  
***From room***  
***Where type=***'***Double***'  
***Group by hotelNo***  
***Having count(type)>2;***

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

//syntactically correct but not supported by SQLite

SELECT \* FROM Room r, hotel h

WHERE r.hotelno = h.hotelno AND h.hotelname = 'Hilton' AND

(r.hotelno,roomNo) NOT IN

(SELECT (h.hotelNo,roomNo) FROM Booking b, Hotel h

WHERE (dateFrom <= CURRENT\_DATE AND

dateTo >= CURRENT\_DATE) AND

b.hotelNo = h.hotelNo AND hotelName = 'Hilton');

//work in SQLite

SELECT \* FROM Room r, hotel h

WHERE r.hotelno = h.hotelno AND h.hotelname = 'Hilton' AND

roomNo NOT IN

(SELECT roomNo FROM Booking b, Hotel h

WHERE (dateFrom <= CURRENT\_DATE AND

dateTo >= CURRENT\_DATE) AND

b.hotelNo = h.hotelNo AND hotelName = 'Hilton');

//This is not correct: (this is showing rooms not being occupied now but has been booked before. If a room never been booked, won’t show up)

SELECT \* FROM Room r, hotel h

WHERE r.hotelno = h.hotelno AND h.hotelname = 'Hilton' AND

roomNo IN

(SELECT roomNo FROM Booking b, Hotel h

WHERE (dateFrom > CURRENT\_DATE OR

dateTo < CURRENT\_DATE) AND

b.hotelNo = h.hotelNo AND hotelName = 'Hilton');

**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 DEPARTMENT, EMPLOYEE

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)

Should be

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

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

If input doesn’t follow all attributes in initial schema, need to specify:

insert into test(id, name) values (2, 'John');

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

Shouldn’t have “values as”

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

Can only delete from 1 table as a time

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

Need to list the table name after update. Shouldn’t have “test.” before id.

Should be: update test set id = 3 where name = 'John';

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

**UPDATE** Employee **SET** salary = salary \* 1.25

**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 ]