

Tutorial 2

Basic SELECT Statement (revision)

SELECT A1, A2, ..., An	(Attributes)
FROM R1, R2, ..., Rn	(Relations/Tables)
WHERE condition	(Selection)

```
SELECT Name
FROM STUDENT
WHERE STUDENTID LIKE "%2017%";
```

```
SELECT *
FROM STUDENT;
```

Hotel(hotelNo, name, address)

Room(roomId, hotelNo, type, price, number)

Booking(roomId, guestNo, dateFrom, dateTo)

Guest(guestNo, name, address)

- Q1. List full details of all hotels.

SELECT * from Hotel;

- Q2. List full details of all hotels in London.

SELECT * from Hotel

WHERE address LIKE '%London%';

Single Column Ordering (REVISION)

ORDER BY and an attribute sorts the values according to the attribute (alphabetically for characters or numerically for numbers)

DESC (for descending) sorts it in reverse order.

```
SELECT staffNo, fName, lName, salary  
FROM Staff  
ORDER BY salary DESC;
```

Hotel(hotelNo, name, address)

Room(roomId, hotelNo, type, price, number)

Booking(roomId, guestNo, dateFrom, dateTo)

Guest(guestNo, name, address)

- Q3. List the names and addresses of all guests in London, alphabetically ordered by name.

SELECT name, address

FROM Guest

WHERE address LIKE '%London%'

ORDER BY name;

Hotel(hotelNo, name, address)

Room(roomId, hotelNo, type, price, number)

Booking(roomId, guestNo, dateFrom, dateTo)

Guest(guestNo, name, address)

- Q4. List all double or family rooms with a price below 40.00 per night, in ascending order of price.

```
SELECT * FROM Room  
WHERE (type = 'double' OR  
       type = 'family') AND  
       price < 40  
ORDER BY price;
```

SELECT Statement – Aggregates (revision)

- COUNT (how many) MIN (minimum) MAX (maximum) AVG (average) SUM (sum)
- Apart from COUNT(*), each function eliminates nulls first and operates only on remaining non-null values.
- Can use DISTINCT before column name to eliminate duplicates.
- DISTINCT has no effect with MIN/MAX, but may have with SUM/AVG.

NULL Search Condition (revision)

Test for or null explicitly using special keyword IS NULL:
(opposite IS NOT NULL)

```
SELECT clientNo, viewDate
FROM Viewing
WHERE propertyNo = "PG4" AND
      comment IS NULL;
```


Hotel(hotelNo, name, address)
Room(roomId, hotelNo, type, price, number)
Booking(roomId, guestNo, dateFrom, dateTo)
Guest(guestNo, name, address)

- Q5. List the bookings for which no dateTo has been specified.

**SELECT * FROM Booking
WHERE dateTo IS NULL;**

- Q6. How many hotels are there?

SELECT COUNT(*) FROM Hotel;

Hotel(hotelNo, name, address)

Room(roomId, hotelNo, type, price, number)

Booking(roomId, guestNo, dateFrom, dateTo)

Guest(guestNo, name, address)

- Q7. What is the average price of all the rooms?

SELECT AVG(price)

FROM Room;

Hotel(hotelNo, name, address)

Room(roomId, hotelNo, type, price, number)

Booking(roomId, guestNo, dateFrom, dateTo)

Guest(guestNo, name, address)

Q8. How many different guests have made bookings for August 2013 (inclusive)?

```
SELECT COUNT(DISTINCT guestNo)  
FROM Booking  
WHERE dateFrom >= '2013-08-01' AND  
       dateTo <= '2013-08-31' ;
```

Simple Join (Revision)

Join two tables by including WHERE and setting columns equal (you can use AND to join more than one column). You can use more than two tables.

```
SELECT c.clientNo, fName, lName,  
       propertyNo, comment  
FROM Client c, Viewing v  
WHERE c.clientNo = v.clientNo;
```

Hotel(hotelNo, name, address)

Room(roomId, hotelNo, type, price, number)

Booking(roomId, guestNo, dateFrom, dateTo)

Guest(guestNo, name, address)

- Q9. List the price and type of all rooms at the Grosvenor Hotel.

SELECT price, type

FROM Hotel h, Room r

WHERE h.hotelNo = r.hotelNo AND

h.name = 'Grosvenor Hotel';

Hotel(hotelNo, name, address)
Room(roomId, hotelNo, type, price, number)
Booking(roomId, guestNo, dateFrom, dateTo)
Guest(guestNo, name, address)

- Q10. List all information for all guests currently staying at the Grosvenor Hotel. (`CURRENT_DATE()` function in MySQL)

```
SELECT g.guestNo, g.name, g.address
FROM Guest g, Booking b, Hotel h, Room r
WHERE h.hotelNo = r.hotelNo AND
      g.guestNo = b.guestNo AND
      r.roomId = b.roomId AND
      h.name = 'Grosvenor Hotel' AND
      dateFrom <= CURRENT_DATE() AND
      dateTo > CURRENT_DATE();
```

Hotel(hotelNo, name, address)
Room(roomId, hotelNo, type, price, number)
Booking(roomId, guestNo, dateFrom, dateTo)
Guest(guestNo, name, address)

- Q11. What is the total income from bookings for people staying in the hotel with hotelNo of 015 today?

```
SELECT SUM(price)  
FROM Booking b, Room r  
WHERE r.roomId = b.roomId AND  
      dateFrom <= CURRENT_DATE() AND  
      dateTo > CURRENT_DATE() AND  
      r.hotelNo = '015';
```

Revision: Nested subquery: use of IN

List properties handled by staff at '163 Main St'.

```
SELECT  propertyNo,    street,    city,    postcode,  
        type, rooms, rent  
FROM PropertyForRent  
WHERE staffNo IN  
      (SELECT staffNo  
        FROM Staff  
        WHERE branchNo =  
              (SELECT branchNo  
                FROM Branch  
                WHERE street = '163 Main St'));
```


Hotel(hotelNo, name, address)

Room(roomId, hotelNo, type, price, number)

Booking(roomId, guestNo, dateFrom, dateTo)

Guest(guestNo, name, address)

- Q12. List information of the rooms that are currently unoccupied. (Hint: find rooms that are occupied currently first)

SELECT *

FROM Room r

WHERE roomId NOT IN

(SELECT roomId FROM Booking b,

WHERE dateFrom <= CURRENT_DATE() AND

dateTo > CURRENT_DATE());

SQL JOIN(Revision)

NATURAL JOIN

JOIN USING

JOIN ON

LEFT JOIN – if left table has no match add NULL

RIGHT JOIN – if right table has no match add NULL

FULL JOIN – if either table has no match add NULL

Hotel(hotelNo, name, address)

Room(roomId, hotelNo, type, price, number)

Booking(roomId, guestNo, dateFrom, dateTo)

Guest(guestNo, name, address)

- Q13. List the guestNo of guests who have booked executive rooms.

```
SELECT guestNo
FROM Booking NATURAL JOIN Room
WHERE type = "executive";
```

```
SELECT guestNo
FROM Booking JOIN Room USING (roomId)
WHERE type = "executive";
```

```
SELECT guestNo
FROM Booking b JOIN Room r ON b.roomId = r.roomId
WHERE type = "executive";
```

Hotel(hotelNo, name, address)

Room(roomId, hotelNo, type, price, number)

Booking(roomId, guestNo, dateFrom, dateTo)

Guest(guestNo, name, address)

- Q14. List all guests and their booking information, display NULL for the booking if the guest has not booked a room.

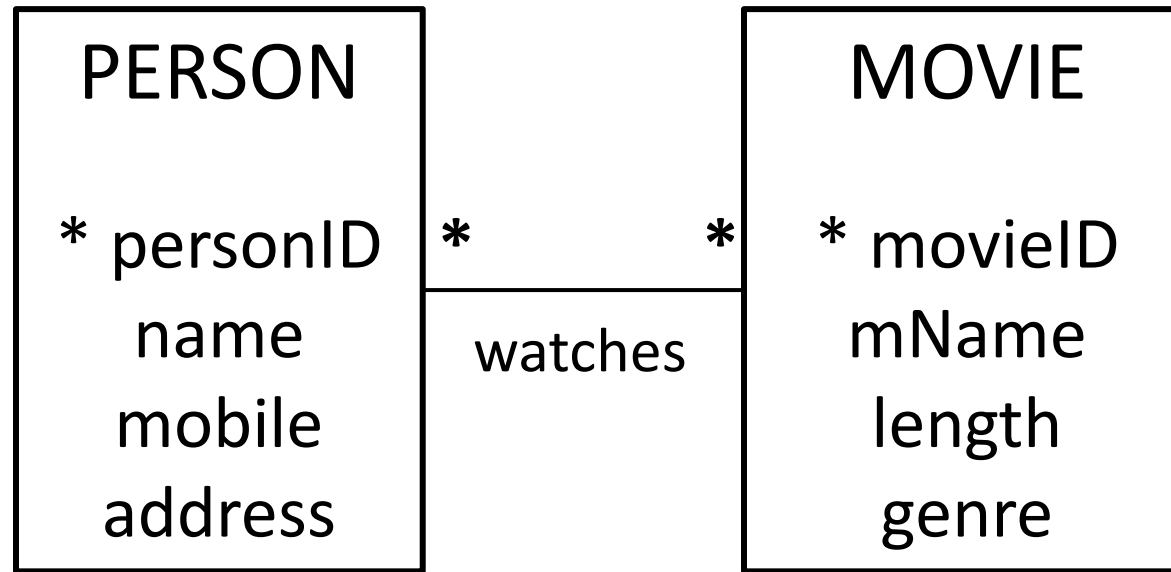
SELECT *

**FROM Guest g LEFT JOIN Booking b ON g.guestNo =
b.guestNo**

SELECT *

**FROM Booking b RIGHT JOIN Guest g ON g.guestNo =
b.guestNo**

Q15. Derive relational schema for the entities and relationships in the ER model below.



If a m:m relationship is not broken down in the ER diagram, we need to add associative entity for the relationship.

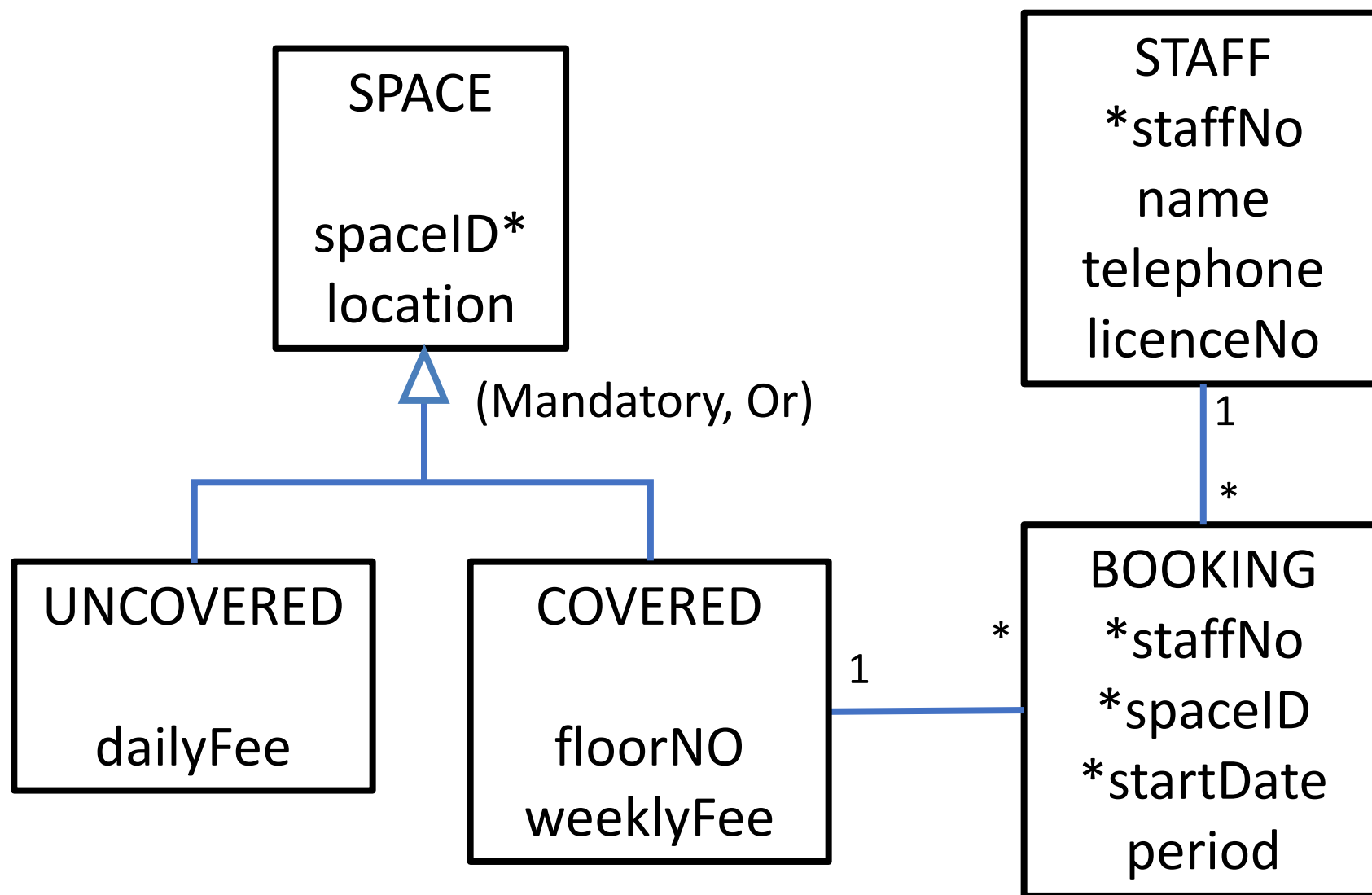
PERSON(personID, name, mobile, address)

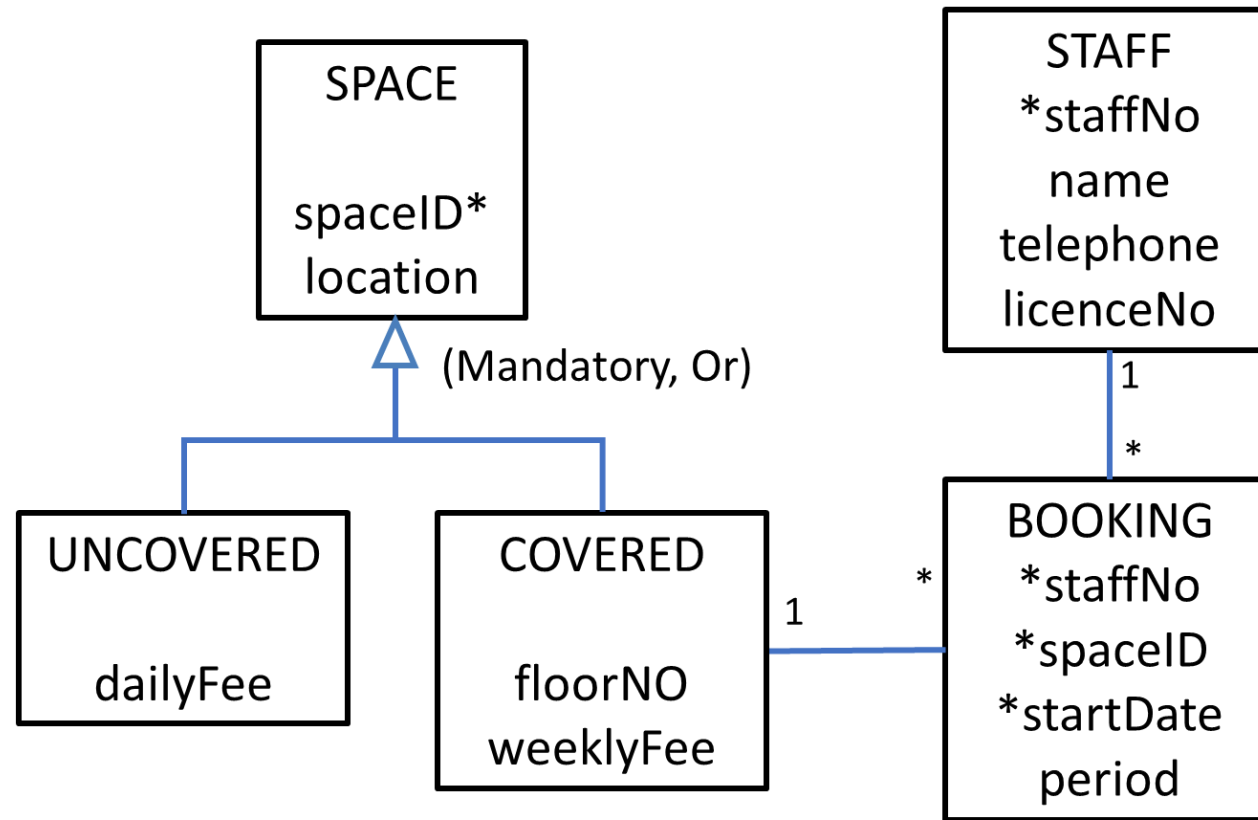
MOVIE (movieID, mName, length, genre)

WATCH (watchRef, personID, movieID, date, time...)

Foreign key in WATCH are personID, movieID

Q16. Derive relational schema for the entities and relationships in the ER model below.





UNCOVERED (spaceID, location, dailyFee)

COVERED (spaceID, location, floorNo, weeklyFee)

STAFF (staffNo, name, telephone, licenceNo)

BOOKING (staffNo, spaceID, startDate, period)