INFO90002 Week 3 Tutorial

Objectives:

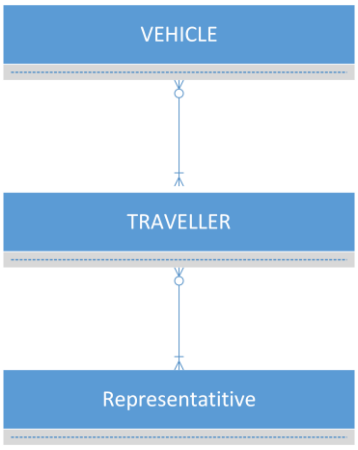
- Review the Case Study (5 minutes)
- Draw a conceptual and logical model (10 minutes)
- Use MySQL modelling tool to build tables (15 minutes)
- Practice writing SQL (20 minutes)

Section 1. Data Modelling

Case Study: TraveLeCar Car Rentals

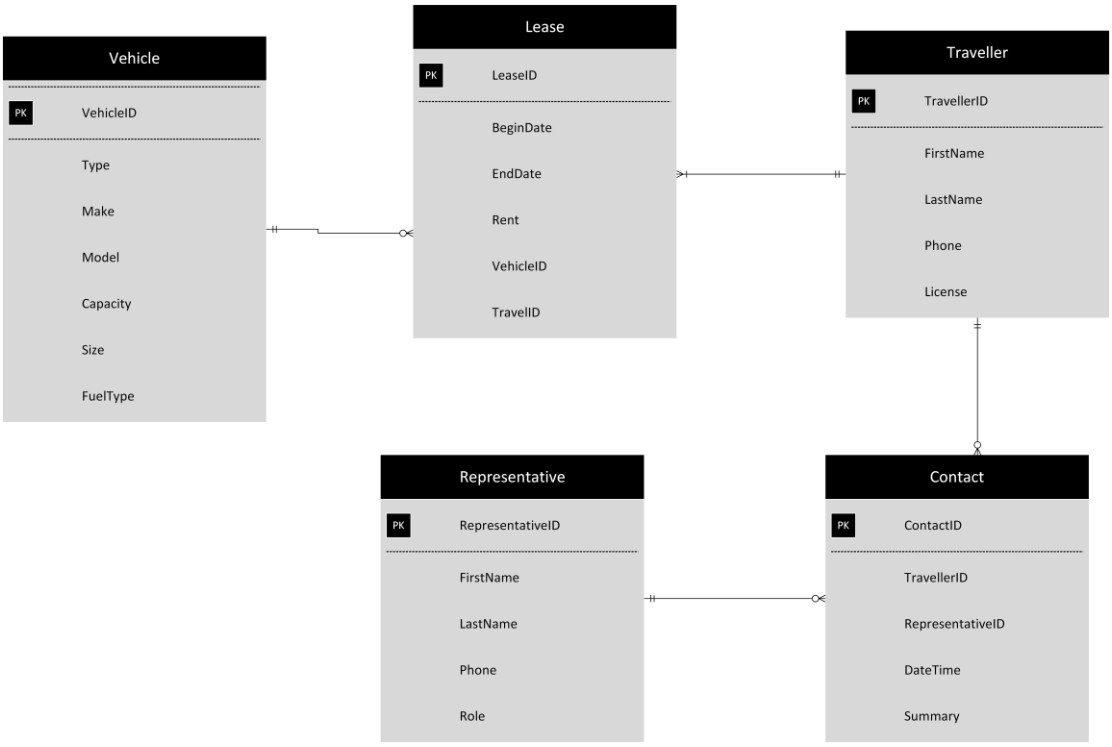
The TraveLeCar corporation leases vehicles to holiday travellers. Travellers can sign a lease from 1 week upto 3 years for a minimum rental unit of a week. About each vehicle TraveLeCar leases it records the type of vehicle (motorcycle, car) the make (Citreon), the model (CV2) , the advertised weekly rent ($250), the number of passengers, size, and fuel type (electric, gas, petrol). TraveLeCar stores the name and phone number and drivers license number of the individual who signs the lease. When a new holiday traveller arrives, we record the date they hired the vehicle and the weekly rent they will be charged. When the lease is over and the traveller returns the car TraveLeCar record the date the vehicle was returned. To manage customer relationships, representatives from TraveLeCar keep track of every contact they have with the holiday traveller during the lease, recording the name of who from TraveLeCar contacted the traveller and the date and time, and up to a 10 word summary of the conversation (e.g “Vehicle Inspection Wed 24th March”). We record the first name, last name, phone number and role each of each TraveLeCar representative.

1.1. Using a pen and paper create a conceptual model from the MiCasaSuCasa case study



Suggested Conceptual model for TraveLeCar in Crows Foot notation

1.2 Using a pen and paper create a logical model for a relational database from your conceptual model



It is important even at the conceptual and logical stages to be consistent in your naming conventions. That is if your entities will be plural or singular (e.g. Employees or Employee) whether you use Camel Caps (aka InitCaps) or all capital letters (e.g EMPLOYEE Employee) whatever you decide to do, be consistent across all your designs and entities.

MySQL Workbench Data Modelling

In this section we will create the tables in MySQL Workbench

1.3 Use MySQL Workbench modelling tool add each table and the key attributes. Be sure to save your model.

MySQL Workbench has three sections: the default window, the modelling window and the migration window.

1.4 Launch MySQL Workbench

When you launch MySQL Workbench you will be in the default window. This is highlighted in the right frame of MySQL Worksheet and circled in red below:

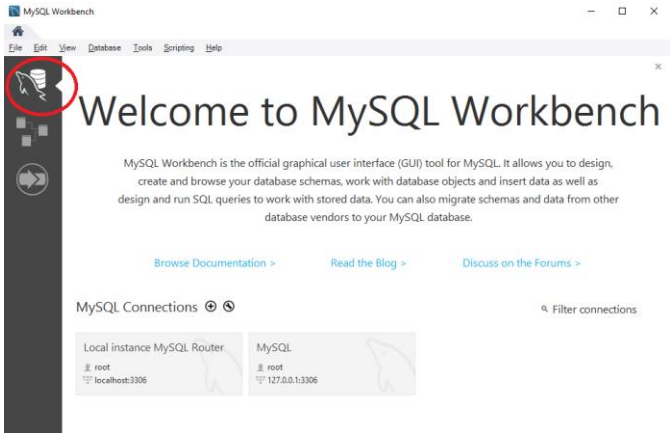


Figure 1. The default MySQL Workbench window

You will need to select the data modelling window

Figure 2: The Modeling icon

This will now be the highlighted window with the pointer:

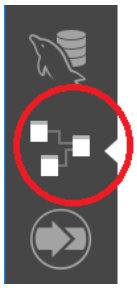


Figure 3: The active window for data modelling

And your Window will have changed to a different view:

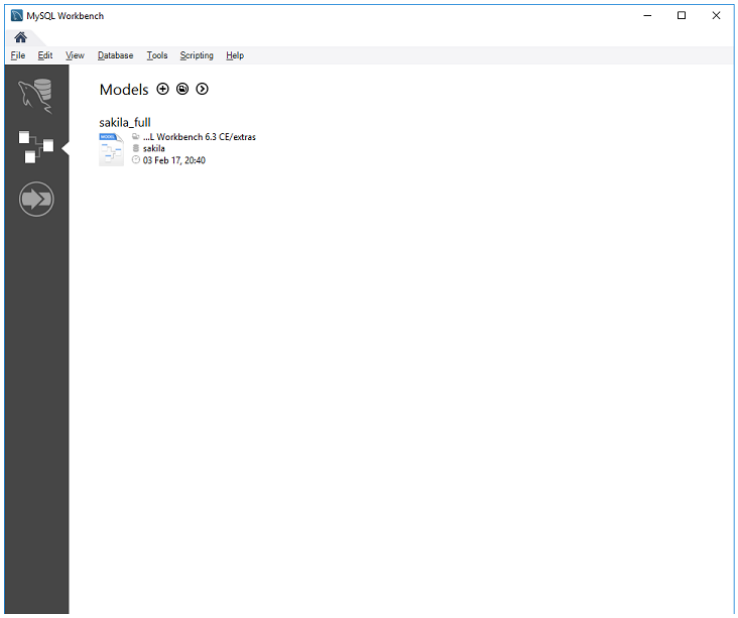


Figure 4: The full screen of the data modelling module

1.5 Click the Add model (+) symbol next to the Model at the top left of the MySQL Window.

This will launch a new modelling window:

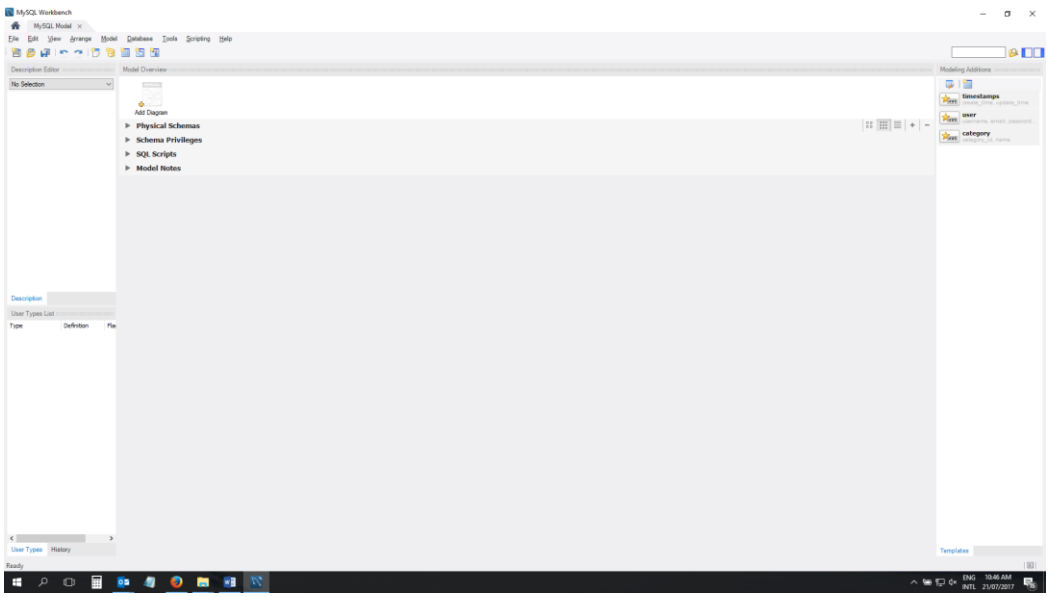


Figure 5: The new modelling window

1.6 Click the “Add Diagram” icon to add a new diagram:

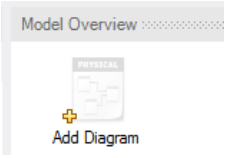


Figure 6: The add diagram icon

This will bring a second tab and a diagram window:

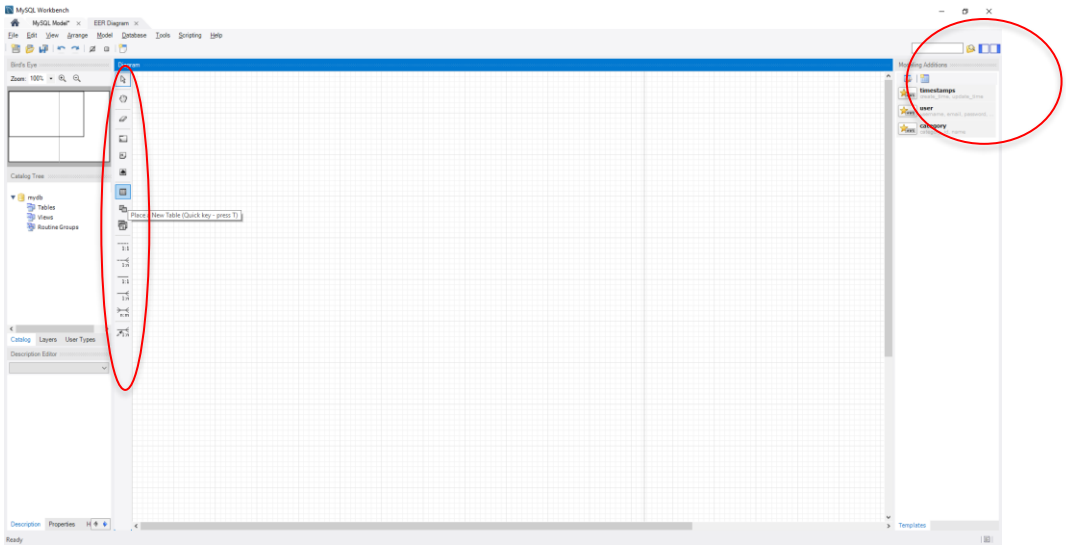


Figure 7: The diagram with modelling tools on the left hand side of the canvas. Note the minimizing of windows icon in the top right hand corner of the canvas

1.7 Make more space for your diagram. Remove the left and right side columns of your modelling canvas:

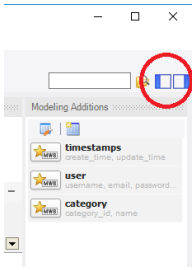


Figure 9: Highlight of the icon to hide the right and left columns of the modelling canvas

Adding tables to your model

1.5 Using your mouse click once on the table icon and drag it to your diagram

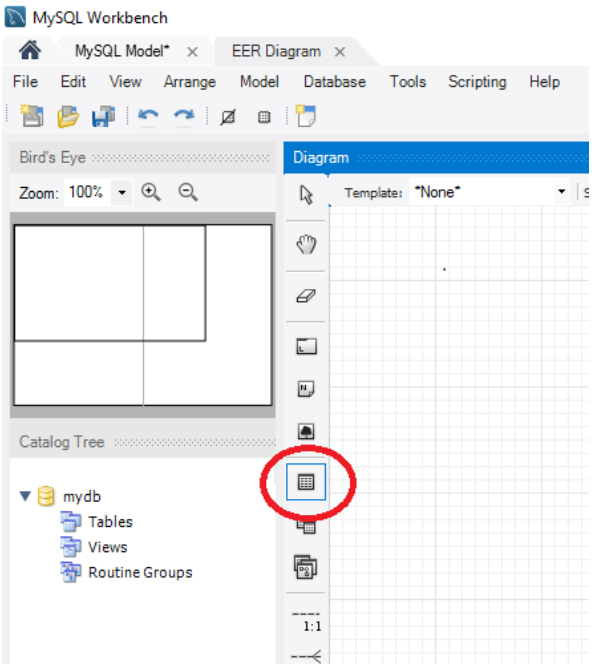


Figure 10 The add table tool

1.6 Drop the table on your diagram

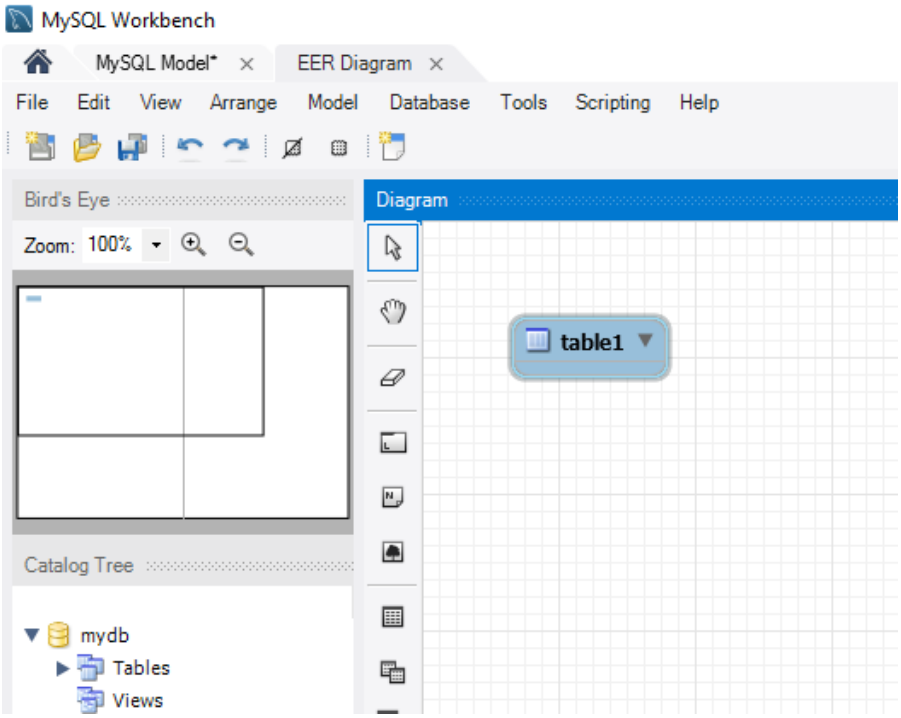


Figure 11 The table has now been placed on the diagram

1.7 Double click the table and the table wizard appears at the bottom of the application

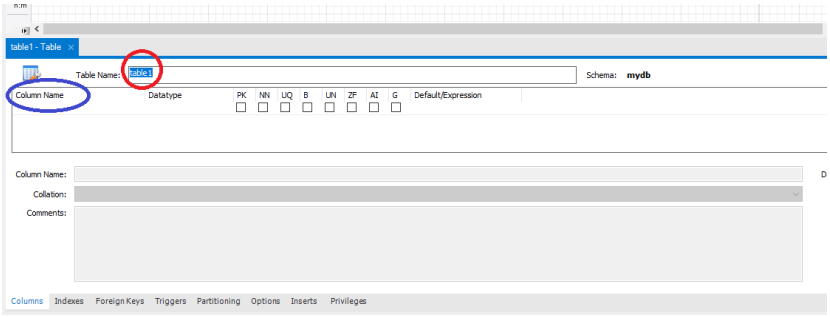


Figure 12 The Table Wizard dialog box

The Table name is highlighted

1.8 Add the Vehicle and Lease tables

Using MySQL Workbench modelling tool add the following 2 tables

Vehicle Table

Column	Data type
VehicleID	INTEGER
TYPE	VARCHAR(45)
MAKE	VARCHAR(45)
MODEL	VARCHAR(45)
CAPACITY	INTEGER
SIZE	INTEGER
FUELTYPE	VARCHAR(45)

Lease Table

Column	Data type
LeaseID	INTEGER
BeginDate	DATE
EndDate	DATE
RENT	DECIMAL(6,2)
VehicleID	INTEGER
TravellerID	INTEGER

1.9 Save your model

END of Week 3 Tutorial

Section 2 SQL

Using MySQL Workbench, connect to your database to the MySQL Server

2.1 Show the catalogue (meta data) about the Department table

```
DESC Department;
```

Hint: if you get an error message or can't find the Department table, check that you ran the Setup Script in week 2 you may also need to tell the database to use the correct schema

```
use <username>;
```

Structure of SQL statements

The structure of a SELECT statement is below:

```
SELECT (select list)
FROM (from list)
WHERE (filtering list) -- optional
GROUP BY (grouping list) -- optional
HAVING (group qualifications) -- optional
```

To select all columns from a table we use the SQL shorthand "*"

To select from the Department table, enter the following SQL:

```
SELECT *
FROM department;
```

DepartmentID	Name	Floor	Phone	ManagerID
1	Management	5	34	1
2	Books	1	81	4
3	Clothes	2	24	4
4	Equipment	3	57	3
5	Furniture	4	14	3
6	Navigation	1	41	3
7	Recreation	2	29	4
8	Accounting	5	35	5
9	Purchasing	5	36	7
10	Personnel	5	37	9
11	Marketing	5	38	2
NULL	NULL	NULL	NULL	NULL

2.2 Type the SQL query to select all rows and columns from the Employee table.

You should see a result like this:

EmployeeID	FirstName	LastName	Salary	DepartmentID	BossID	DateOfBirth
1	Alice	Munro	125000.00	1	NULL	1966-12-14
2	Ned	Kelly	85000.00	11	1	1970-07-16
3	Andrew	Jackson	55000.00	11	2	1958-04-01
4	Clare	Underwood	52000.00	11	2	1982-09-22
5	Todd	Beamer	68000.00	8	1	1965-05-24
6	Nancy	Cartwright	52000.00	8	5	1993-04-11
7	Brier	Patch	73000.00	9	1	1981-10-16
8	Sarah	Fergusson	86000.00	9	7	1978-11-15
9	Sophie	Monk	75000.00	10	1	1986-12-15
10	Sanjay	Patel	45000.00	6	3	1984-01-28
11	Rita	Skeeter	45000.00	2	4	1988-02-22
12	Gigi	Montez	46000.00	3	4	1992-03-20
13	Maggie	Smith	46000.00	3	4	1991-04-29
14	Paul	Innit	41000.00	4	3	1998-06-02
15	James	Mason	45000.00	4	3	1995-07-30
16	Pat	Clarkson	45000.00	5	3	1997-08-28
17	Mark	Zhang	45000.00	7	3	1996-10-01

```
SELECT *
FROM employee;
```

Filtering Results

To select only some columns, we specify the columns we want in the query. Separate each column name with a ",". If you describe the Department table we can see attributes Name and Floor, amongst others.

To select only these two columns in the Department table, enter this SQL:

```
SELECT Name, Floor
FROM department;
```

You should see a result set like this:

Name	Floor
Management	5
Books	1
Clothes	2
Equipment	3
Furniture	4
Navigation	1
Recreation	2
Accounting	5
Purchasing	5
Personnel	5
Marketing	5

This is known as a *projection* to filter the result set by columns.

2.3 Type the SQL query to select the first name, last name and departmentid in the Employee table.

firstname	lastname	departmentid
Alice	Munro	1
Ned	Kelly	11
Andrew	Jackson	11
Clare	Underwood	11
Todd	Beamer	8
Nancy	Cartwright	8
Brier	Patch	9
Sarah	Fergusson	9
Sophie	Monk	10
Sanjay	Patel	6
Rita	Skeeter	2
Gigi	Montez	3
Maggie	Smith	3
Paul	Innit	4
James	Mason	4
Pat	Clarkson	5
Mark	Zhang	7

```
SELECT firstname, lastname, departmentid
FROM department;
```

Until now we have selected **all** the rows in a table. Most times we don't want to retrieve all rows.

In SQL we do this by using a *selection condition* on our query. This is also known as a *selection*. If for example, we wished to list all Departments that are listed on the second floor:

```
SELECT *
FROM department
WHERE floor = 2;
```

DepartmentID	Name	Floor	Phone	ManagerID
3	Clothes	2	24	4
7	Recreation	2	29	4
NULL	NULL	NULL	NULL	NULL

The SQL keyword 'WHERE' lists any horizontal filter we wish to put on the query.

How do we find out all department names that start with M? To do this we use the wildcard '%'. % stands for any character or characters. When you use % you need the SQL word LIKE. To display all departments that start with M we type:

```
SELECT *
FROM department
WHERE Name LIKE 'M%';
```

DepartmentID	Name	Floor	Phone	ManagerID
1	Management	5	34	1
11	Marketing	5	38	2
NULL	NULL	NULL	NULL	NULL

2.4 Type the SQL query to return the first and last names and department id of all employees who earn less than $55000.

```
SELECT firstname, lastname, departmentid
FROM employee
WHERE Salary < 55000;
```

Multiple Conditions

Sometimes we may need to filter the result set by having more than one condition met. For example, if we wish to list all the Departments that start with M and whose manager ID is 1

```
SELECT *
FROM department
WHERE Name like 'M%'
AND ManagerID = 1;
```

Both conditions must be true, and in this case only 1 row is returned:

DepartmentID	Name	Floor	Phone	ManagerID
1	Management	5	34	1
NULL	NULL	NULL	NULL	NULL

However if we change the AND to OR the result set changes. Two rows are returned.

When we use an OR condition, only one condition need be true for a row to be returned.

```
SELECT *
FROM department
WHERE Name like 'M%'
OR ManagerID = 1;
```

The query lists all departments starting with M, as well as all departments where the ManagerID is equal to 1.

DepartmentID	Name	Floor	Phone	ManagerID
1	Management	5	34	1
11	Marketing	5	38	2
NULL	NULL	NULL	NULL	NULL

2.5 In MySQL Workbench show the metadata about the Department and Employee tables

```
DESC Department;
DESC Employee;
```

Mathematical operators

To select all departments that are above the first floor we would type

```
SELECT *
FROM department
WHERE Floor > 1;
```

DepartmentID	Name	Floor	Phone	ManagerID
1	Management	5	34	1
3	Clothes	2	24	4
4	Equipment	3	57	3
5	Furniture	4	14	3
7	Recreation	2	29	4
8	Accounting	5	35	5
9	Purchasing	5	36	7
10	Personnel	5	37	9
11	Marketing	5	38	2
NULL	NULL	NULL	NULL	NULL

Or find out which departments are NOT on the fifth floor

```
SELECT Name, Floor
FROM department
WHERE Floor != 5;
```

OR

```
SELECT Name, Floor
FROM department
WHERE Floor <> 5;
```

Note that != and <> both mean 'Not Equal To'

Name	Floor
Books	1
Clothes	2
Equipment	3
Furniture	4
Navigation	1
Recreation	2

Order By

We can order the result set by any column (it does not have to be in the SELECT clause). The ORDER BY forces the result set to be ordered by the values of one or more columns.

```
SELECT Name, Floor
FROM department
WHERE Floor != 5
ORDER BY Floor;
```

Name	Floor
Books	1
Navigation	1
Clothes	2
Recreation	2
Equipment	3
Furniture	4

The default sort order is from the smallest value to largest (1-10 or A-Z). You can explicitly state this by typing ASC (short for Ascending order). To sort from largest value to smallest you would enter DESC (short for Descending order).

```
SELECT *
FROM department
ORDER BY Floor DESC;
```

DepartmentID	Name	Floor	Phone	ManagerID
1	Management	5	34	1
8	Accounting	5	35	5
9	Purchasing	5	36	7
10	Personnel	5	37	9
11	Marketing	5	38	2
5	Furniture	4	14	3
4	Equipment	3	57	3
3	Clothes	2	24	4
7	Recreation	2	29	4
2	Books	1	81	4
6	Navigation	1	41	3
NULL	NULL	NULL	NULL	NULL

You can order by more than one column and in different order for each column:

```
SELECT *
FROM department
ORDER BY Floor DESC, DepartmentID ASC;
```

DepartmentID	Name	Floor	Phone	ManagerID
1	Management	5	34	1
8	Accounting	5	35	5
9	Purchasing	5	36	7
10	Personnel	5	37	9
11	Marketing	5	38	2
5	Furniture	4	14	3
4	Equipment	3	57	3
3	Clothes	2	24	4
7	Recreation	2	29	4
2	Books	1	81	4
6	Navigation	1	41	3
NULL	NULL	NULL	NULL	NULL

2.6 Type the SQL query that lists the first name, last name of all employees who work in DepartmentID 11 AND who earn greater than 55000.

```
SELECT firstname, lastname
FROM employee
WHERE departmentid=11
AND salary > 55000;
```

firstname	lastname	departmentid	salary
Ned	Kelly	11	85000.00

Then change the AND to OR and note the difference in the result set.

Notice Clare Underwood's department and salary

firstname	lastname	departmentid	salary
Alice	Munro	1	125000.00
Ned	Kelly	11	85000.00
Andrew	Jackson	11	55000.00
Clare	Underwood	11	52000.00
Todd	Beamer	8	68000.00
Brier	Patch	9	73000.00
Sarah	Fergusson	9	86000.00
Sophie	Monk	10	75000.00

2.7 Type the SQL query that returns all employees who earn 45,000 or more. Order the results from highest earner to lowest

```
SELECT firstname, lastname, departmentid
FROM employee
WHERE salary >= 45000
ORDER by salary DESC;
```

Your result set should look like this

firstname	lastname	salary	departmentid
Alice	Munro	125000.00	1
Sarah	Fergusson	86000.00	9
Ned	Kelly	85000.00	11
Sophie	Monk	75000.00	10
Brier	Patch	73000.00	9
Todd	Beamer	68000.00	8
Andrew	Jackson	55000.00	11
Clare	Underwood	52000.00	11
Nancy	Cartwright	52000.00	8
Gigi	Montez	46000.00	3
Maggie	Smith	46000.00	3
Sanjay	Patel	45000.00	6
Rita	Skeeter	45000.00	2
James	Mason	45000.00	4
Pat	Clarkson	45000.00	5
Mark	Zhang	45000.00	7

2.8 Type the SQL query that returns all rows and columns in the employee table. Order the result set by departmentid then alphabetically by employee's lastname.

```
SELECT *
FROM employee
ORDER BY departmentid, lastname;
```

EmployeeID	FirstName	LastName	Salary	DepartmentID	BossID	DateOfBirth
1	Alice	Munro	125000.00	1	NULL	1966-12-14
11	Rita	Skeeter	45000.00	2	4	1988-02-22
12	Gigi	Montez	46000.00	3	4	1992-03-20
13	Maggie	Smith	46000.00	3	4	1991-04-29
14	Paul	Innit	41000.00	4	3	1998-06-02
15	James	Mason	45000.00	4	3	1995-07-30
16	Pat	Clarkson	45000.00	5	3	1997-08-28
10	Saniav	Patel	45000.00	6	3	1984-01-28
17	Mark	Zhang	45000.00	7	3	1996-10-01
5	Todd	Beamer	68000.00	8	1	1965-05-24
6	Nancy	Cartwright	52000.00	8	5	1993-04-11
8	Sarah	Fergusson	86000.00	9	7	1978-11-15
7	Brier	Patch	73000.00	9	1	1981-10-16
9	Sophie	Monk	75000.00	10	1	1986-12-15
3	Andrew	Jackson	55000.00	11	2	1958-04-01
2	Ned	Kelly	85000.00	11	1	1970-07-16
4	Clare	Underwood	52000.00	11	2	1982-09-22

Limit

We can limit the number of rows in the result set by using the word LIMIT and specifying an integer after the LIMIT word.

```
SELECT Name
FROM department
WHERE Floor = 5
ORDER BY Name ASC
LIMIT 2;
```

Name
Accounting
Management

2.9 Type the above query and note the two rows returned. Change the ORDER BY from ASC to DESC and rerun the query. Is the result set different? If so, why are they different? [Hint: remove the LIMIT.

2.10 Type the SQL query :that returns the first name, last name and salary of the five highest salary earners across the whole Department store.

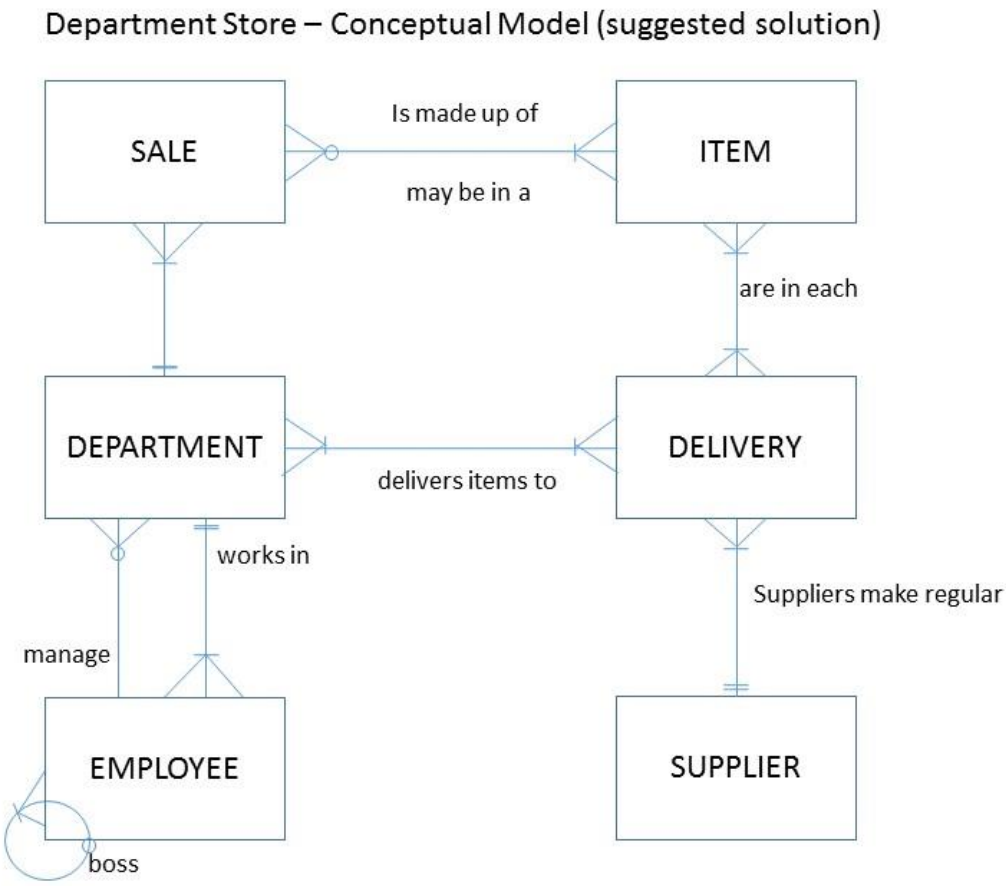
```
SELECT firstname, lastname, salary
FROM employee
ORDER BY salary DESC
LIMIT 5:
```

Your result set should look like this

	firstname	lastname	salary
▶	Alice	Munro	125000.00
	Sarah	Fergusson	86000.00
	Ned	Kelly	85000.00
	Sophie	Monk	75000.00
	Brier	Patch	73000.00

END OF Week 3 LAB

Appendix – Department Store conceptual design



Appendix New department Store Physical ER Model

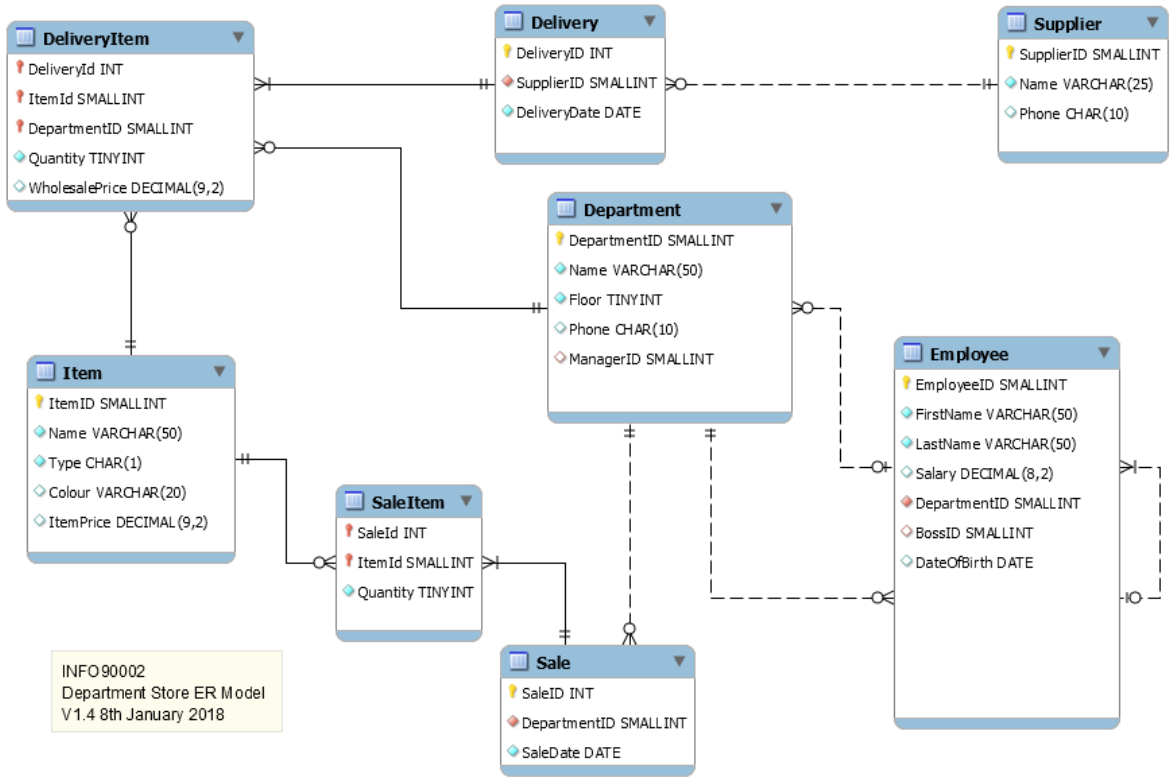


Figure A1: The new department store schema