

Data Retrieval

Natalia Criado

Email: natalia.criado@kcl.ac.uk

Review

Data Retrieval

Introduction

Retrieval

Select

Where

Sorting

Aggregation

Grouping

Having

Tutorial

Exercises

Conclusion

Suggested
Readings

In relational model terminology, table is considered as

A range

B domain

C relation

D tuple

Review

Data Retrieval

Introduction

Retrieval

Select

Where

Sorting

Aggregation

Grouping

Having

Tutorial Exercises

Conclusion

Suggested
Readings

When primary key is null of new tuple then constraint violated is

- A null integrity constraint
- B primary integrity constraint
- C secondary integrity constraint
- D entity integrity constraint

Session Objectives

Data Retrieval

Introduction

Retrieval
Select
Where
Sorting
Aggregation
Grouping
Having

Tutorial Exercises

Conclusion

Suggested
Readings

In this session, you will learn:

- Structured Query Language (SQL)
- SELECT query
 - Select
 - Where
 - Sorting
 - Aggregation
 - Grouping
 - Having

Introduction

Data
Retrieval

Introduction

Retrieval
Select
Where
Sorting
Aggregation
Grouping
Having

Tutorial
Exercises

Conclusion
Suggested
Readings

SQL is a database language that:

- Allows you to create database and table structures, to perform data management tasks and to perform complex queries designed to transform the raw data into useful information.
- It is portable, it is a de facto standard SQL

SQL

Data Retrieval

Introduction

Retrieval
Select
Where
Sorting
Aggregation
Grouping
Having

Tutorial Exercises

Conclusion Suggested Readings

SQL functions fit into two broad categories:

- Data Definition Language (DDL): SQL includes commands to create database objects such as tables, indexes, and views, as well as commands to define access rights to those database objects
- It is a data manipulation language (DML): SQL includes commands to insert, update, delete, and retrieve data within the database tables

Data Manipulation Language (DML)

Data
Retrieval

Introduction

Retrieval
Select
Where
Sorting
Aggregation
Grouping
Having

Tutorial
Exercises

Conclusion
Suggested
Readings

DML allows to retrieve and update data:

- SELECT statement retrieves data
- INSERT, UPDATE, DELETE statements update data (next week)

Writing SQL

Data Retrieval

Introduction

Retrieval
Select
Where
Sorting
Aggregation
Grouping
Having

Tutorial Exercises

Conclusion Suggested Readings

SQL statement consists of *reserved words* and *user-defined words*:

- *Reserved words* are a fixed part of SQL and must be spelt exactly as required
- *User-defined words* are made up by user and represent names of various database objects such as relations, columns, views

Literals in SQL

Data
Retrieval

Introduction

Retrieval
Select
Where
Sorting
Aggregation
Grouping
Having

Tutorial
Exercises

Conclusion
Suggested
Readings

- Literals are constants that are used in SQL statements
- There are different forms of literals for every data type supported by SQL

Writing Literals

Data Retrieval

Introduction

Retrieval
Select
Where
Sorting
Aggregation
Grouping
Having

Tutorial Exercises

Conclusion

Suggested Readings

A literal is a data value commonly used in variable assignments or comparisons:

- *Numeric Literals*: Integers are represented as a sequence of digits. Floats use . as a decimal separator
- *String Literals*: is a sequence of bytes or characters, enclosed within either single quote (') or double quote (") characters
- *Date and Time Literals*: Date and time values can be represented in several formats, such as quoted strings or as numbers, depending on the exact type of the value and other factors
- *Boolean Literals*: The constants TRUE and FALSE evaluate to 1 and 0, respectively. The constant names can be written in any lettercase.

Syntax Notation

Data
Retrieval

Introduction

Retrieval
Select
Where
Sorting
Aggregation
Grouping
Having

Tutorial
Exercises

Conclusion
Suggested
Readings

- Upper-case letters represent reserved words
- | indicates a choice among alternatives.
- Curly braces { indicate a required element.
- Square brackets [indicate an optional element.
- ... indicates optional repetition (0 or more).

Database Example

Data Retrieval

Introduction

Retrieval
Select
Where
Sorting
Aggregation
Grouping
Having

Tutorial Exercises

Conclusion Suggested Readings

We are going to illustrate SQL statements using the following database of a letting agency:

Branch (branchNo, street, city, postcode)

Staff (staffNo, fName, lName, position, sex, DOB, salary, branchNo)

PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent,
ownerNo, staffNo, branchNo)

Data Retrieval

Data Retrieval

Introduction

Retrieval

Select

Where

Sorting

Aggregation

Grouping

Having

Tutorial

Exercises

Conclusion

Suggested
Readings

- The purpose of the `SELECT` statement is to retrieve and display data from one or more database tables

Select Statement

Data Retrieval

Introduction

Retrieval

Select

Where

Sorting

Aggregation

Grouping

Having

Tutorial

Exercises

Conclusion

Suggested
Readings

Select Statement

```
SELECT [DISTINCT | ALL]
      { * | [columnExpression [AS newName]] [, ...] }
FROM  tableName [alias] [, ...]
[WHERE condition]
[GROUP BY columnList] [HAVING condition]
[ORDER BY columnList]
```

- Order of the clauses cannot be changed
- Only SELECT and FROM are mandatory

Select Statement Execution

Data Retrieval

Introduction

Retrieval

- Select
- Where
- Sorting
- Aggregation
- Grouping
- Having

Tutorial Exercises

Conclusion

- Suggested
Readings

FROM	Specifies table(s) to be used
WHERE	Filters rows
GROUP BY	Forms groups of rows with same column value
HAVING	Filters groups subject to some condition
SELECT	Specifies which columns are to appear in output
ORDER BY	Specifies the order of the output

Simple SELECT

Data Retrieval

Introduction

Retrieval
Select
Where
Sorting
Aggregation
Grouping
Having

Tutorial Exercises

Conclusion Suggested Readings

```
SELECT [DISTINCT | ALL]  
      { * | [columnExpression [AS newName]] [, ...] }  
FROM tableName [alias] [, ...]
```

- Can use * as an abbreviation for all columns
- Use DISTINCT to eliminate duplicates
- To name column, use AS clause

Simple SELECT: Example

Data Retrieval

Introduction

Retrieval
Select
Where
Sorting
Aggregation
Grouping
Having

Tutorial Exercises

Conclusion

Suggested Readings

Branch (branchNo, street, city, postcode)
Staff (staffNo, fName, lName, position, sex, DOB, salary, branchNo)
PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo, staffNo, branchNo)

List full details of all staff

```
SELECT *  
FROM Staff;
```

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000.00	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000.00	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000.00	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000.00	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000.00	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000.00	B005

Simple SELECT: Exercise

Data Retrieval

Introduction

Retrieval

Select

Where

Sorting

Aggregation

Grouping

Having

Tutorial

Exercises

Conclusion

Suggested Readings

Branch (branchNo, street, city, postcode)
Staff (staffNo, fName, lName, position, sex, DOB, salary, branchNo)
PropertyForRent (propertyNo, street, city, postcode, type, rooms,
rent, ownerNo, staffNo, branchNo)

Produce a list of salaries for all staff, showing only the staff number, the first and last names, and the salary details.

Simple SELECT: Exercise

Data Retrieval

Introduction

Retrieval

Select

Where

Sorting

Aggregation

Grouping

Having

Tutorial Exercises

Conclusion

Suggested Readings

Produce a list of salaries for all staff, showing only the staff number, the first and last names, and the salary details.

```
SELECT staffNo, fName, lName, salary  
FROM Staff;
```

staffNo	fName	lName	salary
SL21	John	White	30000.00
SG37	Ann	Beech	12000.00
SG14	David	Ford	18000.00
SA9	Mary	Howe	9000.00
SG5	Susan	Brand	24000.00
SL41	Julie	Lee	9000.00

Row selection (WHERE clause)

Data Retrieval

Introduction

Retrieval

Select

Where

Sorting

Aggregation

Grouping

Having

Tutorial

Exercises

Conclusion

Suggested Readings

```
SELECT [DISTINCT | ALL]
        { * | [columnExpression [AS newName]] [, ...] }
FROM tableName [alias] [, ...]
```

[WHERE condition]

Comparison	Compare the value of one expression to the value of another expression
BETWEEN/NOT BETWEEN	Test whether the value of an expression falls within a specified range of values
IN/NOT IN	Test whether the value of an expression equals one of a set of values
LIKE/NOT LIKE	Test whether a string matches a specified pattern
IS NULL/NOT IS NULL	Test whether a column has a null value

Comparison

Data Retrieval

Introduction

Retrieval

Select

Where

Sorting

Aggregation

Grouping

Having

Tutorial

Exercises

Conclusion

Suggested Readings

```
SELECT [DISTINCT | ALL]
        { * | [columnExpression [AS newName]] [, ...] }
FROM tableName [alias] [, ...]

[WHERE condition]
```

- The following simple comparison operators are available:
=, <>, <, ≤, >, ≥
- More complex predicates can be generated using the logical operators AND, OR and NOT

Comparison: Example

Data Retrieval

Introduction

Retrieval

Select

Where

Sorting

Aggregation

Grouping

Having

Tutorial

Exercises

Conclusion

Suggested Readings

branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B003	163 Main St	Glasgow	G11 9QX
B002	56 Clover Dr	London	NW10 6EU

```
SELECT branchNo,street,city,postcode
FROM branch
WHERE city='London' OR city='Glasgow'
```

- What does this query?

BETWEEN / NOT BETWEEN

Data Retrieval

Introduction

Retrieval

Select

Where

Sorting

Aggregation

Grouping

Having

Tutorial

Exercises

Conclusion

Suggested Readings

```
SELECT [DISTINCT | ALL]
      { * | [columnExpression [AS newName]] [, ...] }
FROM tableName [alias] [, ...]
[WHERE condition]
```

- The BETWEEN / NOT BETWEEN is a simpler way to express a search condition when considering a range of values
- They do not add much expressibility since that can be expressed using two comparison tests

Range: Exercise

Data Retrieval

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000.00	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000.00	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000.00	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000.00	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000.00	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000.00	B005

```
SELECT staffNo, fName, lName, position, salary
FROM Staff
WHERE salary BETWEEN 20000 AND 30000
```

- Express this select without using a range operator

IN / NOT IN

Data Retrieval

Introduction

Retrieval

Select

Where

Sorting

Aggregation

Grouping

Having

Tutorial

Exercises

Conclusion

Suggested Readings

```
SELECT [DISTINCT | ALL]
      { * | [columnExpression [AS newName]] [, ...] }
FROM tableName [alias] [, ...]
[WHERE condition]
```

- The IN / NOT IN tests whether a data value matches one of a list of values
- This list of values can be calculated dynamically with a select (subquery)

Membership: Example

Data Retrieval

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000.00	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000.00	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000.00	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000.00	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000.00	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000.00	B005

```
SELECT staffNo, fName, lName, position
FROM Staff
WHERE position IN ('Manager', 'Supervisor')
```

- What does this query?

LIKE / NOT LIKE

Data Retrieval

Introduction

Retrieval
Select
Where
Sorting
Aggregation
Grouping
Having

Tutorial Exercises

Conclusion

Suggested
Readings

```
SELECT [DISTINCT | ALL]
        { * | [columnExpression [AS newName]] [, ...] }
FROM tableName [alias] [, ...]
[WHERE condition]
```

- LIKE / NOT LIKE test whether a string matches a specified pattern:
 - 1 The % percent character represents any sequence of zero or more characters
 - 2 The _ underscore character represents any single character
 - 3 If you want to search for strings with the symbols % or _ you need to escape them (i.e., using \% or _)

Pattern: Example

Data Retrieval

Introduction

Retrieval

Select

Where

Sorting

Aggregation

Grouping

Having

Tutorial

Exercises

Conclusion

Suggested

Readings

OWNERNO	FNAME	LNAME	ADDRESS	TELNO
-----	-----	-----	-----	-----
C046	Joe	Keogh	2 Fergus Dr, Aberdeen AB2 7SX	01224-861212
C087	Carol	<u>Farrel</u>	6 <u>Achray St</u> , Glasgow G32 9DX	0141-357-7419
C040	Tina	Murphy	63 Well St, Glasgow G42	0141-943-1728
C093	Tony	Shaw	12 Park Pl, Glasgow G4 0QR	0141-225-7025

```
SELECT ownerNo, fName, lName, address, telNo
FROM PrivateOwner
WHERE address LIKE '%Glasgow%'
```

- What does this query?

IS NULL / NOT IS NULL

Data Retrieval

Introduction

Retrieval
Select
Where
Sorting
Aggregation
Grouping
Having

Tutorial Exercises

Conclusion

Suggested
Readings

```
SELECT [DISTINCT | ALL]
        { * | [columnExpression [AS newName]] [, ...] }
FROM tableName [alias] [, ...]
[WHERE condition]
```

- IS NULL / NOT IS NULL test whether a column has a null value

Pattern: Example

Data Retrieval

Viewing Table

CLIENTNO	PROP	VIEWDATE	COMMNT
-----	----	-----	-----
CR56	PA14	24-MAY-04	too small
CR76	PG4	20-APR-04	too remote
CR56	PG4	26-MAY-04	
CR62	PA14	14-MAY-04	no dining room
CR56	PG36	28-APR-04	

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

- What does this query?

Row sorting (ORDER BY clause)

Data Retrieval

Introduction

Retrieval

Select

Where

Sorting

Aggregation

Grouping

Having

Tutorial

Exercises

Conclusion

Suggested Readings

```
SELECT [DISTINCT | ALL]
      { * | [columnExpression [AS newName]] [, ...] }
FROM tableName [alias] [, ...]
[WHERE condition]
[ORDER BY columnList]
```

The ORDER BY clause:

- Consists of a list of column identifiers that the result is to be sorted on, separated by commas
- Allows the retrieved rows to be ordered in ascending (ASC) or descending (DESC) order
- ASC is the default ordering option:

ORDER BY type is the same as ORDER by type ASC

Sorting: Exercise

Data Retrieval

Introduction

Retrieval

Select

Where

Sorting

Aggregation

Grouping

Having

Tutorial Exercises

Conclusion

Suggested Readings

The following query

```
SELECT propertyNo, type, rooms, rent
FROM PropertyForRent
ORDER BY type
```

produces:

propertyNo	type	rooms	rent
PL94	Flat	4	400
PG4	Flat	3	350
PG36	Flat	3	375
PG16	Flat	4	450
PA14	House	6	650
PG21	House	5	600

- Write an SQL sentence that arranges these four flats in this list in order of rent

Sorting: Exercise

Data Retrieval

Introduction

Retrieval
Select
Where
Sorting
Aggregation
Grouping
Having

Tutorial Exercises

Conclusion Suggested Readings

Write an SQL sentence that arranges these four flats in this list in order of rent

```
SELECT propertyNo, type, rooms, rent  
FROM PropertyForRent  
ORDER BY type,rent DESC
```

propertyNo	type	rooms	rent
PG16	Flat	4	450
PL94	Flat	4	400
PG36	Flat	3	375
PG4	Flat	3	350
PA14	House	6	650
PG21	House	5	600

SQL Aggregate Functions

Data Retrieval

Introduction

Retrieval

Select

Where

Sorting

Aggregation

Grouping

Having

Tutorial

Exercises

Conclusion

Suggested Readings

To perform some form of summation or aggregation of data

- There are eight main aggregate functions:
 - 1** COUNT returns number of values in specified column
 - 2** COUNT (*) counts all rows of a table, regardless of nulls or duplicate values
 - 3** SUM returns sum of values in specified column
 - 4** AVG returns average of values in specified column
 - 5** STDDEV returns standard deviation of values in specified column
 - 6** VARIANCE returns variance of values in specified column
 - 7** MIN returns smallest value in specified column
 - 8** MAX returns largest value in specified column
- COUNT, MIN, and MAX apply to numeric and non-numeric fields, but SUM, STDDEV, VARIANCE and AVG may be used on numeric fields
- DISTINCT can be used to eliminate duplicates

Aggregate: Example

Data Retrieval

Introduction

Retrieval

Select

Where

Sorting

Aggregation

Grouping

Having

Tutorial

Exercises

Conclusion

Suggested Readings

The following query

```
SELECT COUNT(DISTINCT staffNo) AS myCount
FROM Staff
WHERE DOB BETWEEN '01/01/1960'
                AND '31/12/1960'
```

- What does this query?

Aggregate: Example

Data Retrieval

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000.00	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000.00	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000.00	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000.00	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000.00	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000.00	B005

```
SELECT COUNT (staffNo), SUM(salary)
FROM staff
WHERE position='manager'
```

- What does this query?

When Aggregate Functions Can Be Used

Data Retrieval

Introduction

Retrieval

Select

Where

Sorting

Aggregation

Grouping

Having

Tutorial

Exercises

Conclusion

Suggested Readings

- Aggregation functions can only be used in the SELECT list and in the HAVING clause:
 - If the SELECT contains an aggregate function and no GROUP BY clause is being used, then no item in the SELECT list can include any reference to a column unless that column is in the aggregate function

Aggregation: Exercise I

Data Retrieval

Introduction

Retrieval

Select

Where

Sorting

Aggregation

Grouping

Having

Tutorial Exercises

Conclusion

Suggested
Readings

Branch (branchNo, street, city, postcode)
Staff (staffNo, fName, lName, position, sex, DOB, salary, branchNo)
PropertyForRent (propertyNo, street, city, postcode, type, rooms,
rent, ownerNo, staffNo, branchNo)

How many properties cost more than 350 per month to rent?

Aggregation: Exercise II

Data Retrieval

Introduction

Retrieval

Select

Where

Sorting

Aggregation

Grouping

Having

Tutorial Exercises

Conclusion

Suggested
Readings

Branch (branchNo, street, city, postcode)
Staff (staffNo, fName, lName, position, sex, DOB, salary, branchNo)
PropertyForRent (propertyNo, street, city, postcode, type, rooms,
rent, ownerNo, staffNo, branchNo)

Find the minimum, maximum, and average staff salary

Grouping

Data Retrieval

Introduction

Retrieval
Select
Where
Sorting
Aggregation
Grouping
Having

Tutorial Exercises

Conclusion Suggested Readings

GROUP BY `columnList`

- The GROUP BY clause forms groups of rows with same column name (get sub-totals)
- SELECT and GROUP BY closely integrated: each item in SELECT list must be single-valued per group, and SELECT clause may only contain:
 - All column names in SELECT list must appear in GROUP BY clause unless name is used only in an aggregate function
- If WHERE is used with GROUP BY, WHERE is applied first, then groups are formed from remaining rows satisfying predicate

GROUP BY: Example

Data Retrieval

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000.00	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000.00	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000.00	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000.00	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000.00	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000.00	B005

```
SELECT branchNo, COUNT(staffNo) AS myCount,  
       SUM(salary) AS mySum  
FROM Staff  
GROUP BY branchNo;
```

- What does this query?

GROUP BY: Example

Data Retrieval

```
SELECT branchNo, COUNT(staffNo) AS myCount, SUM(salary) AS mySum
FROM Staff
GROUP BY branchNo;
```

branchNo	staffNo	salary		COUNT(staffNo)	SUM(salary)
B003	SG37	12000.00	}	3	54000.00
B003	SG14	18000.00			
B003	SG5	24000.00			
B005	SL21	30000.00	}	2	39000.00
B005	SL41	9000.00			
B007	SA9	9000.00	}	1	9000.00

- 1 SQL divides the staff into groups according to their respective branch numbers
- 2 For each group, SQL computes the number of staff members and calculates the sum of the values in the salary column to get the total of their salaries
- 3 Finally, the result is sorted in ascending order of branch number, branchNo

GROUP BY: Example

Data Retrieval

Introduction

Retrieval
Select
Where
Sorting
Aggregation
Grouping
Having

Tutorial Exercises

Conclusion

Suggested
Readings

```
SELECT branchNo, COUNT(staffNo) AS myCount,  
       SUM(salary) AS mySum  
FROM Staff  
GROUP BY branchNo;
```

branchNo	myCount	mySum
B003	3	54000.00
B005	2	39000.00
B007	1	9000.00

Group By: Exercise I

Data Retrieval

Introduction

Retrieval

Select

Where

Sorting

Aggregation

Grouping

Having

Tutorial

Exercises

Conclusion

Suggested
Readings

Branch (branchNo, street, city, postcode)
Staff (staffNo, fName, lName, position, sex, DOB, salary, branchNo)
PropertyForRent (propertyNo, street, city, postcode, type, rooms,
rent, ownerNo, staffNo, branchNo)

List the number of properties managed by each branch

Having

Data Retrieval

Introduction

Retrieval
Select
Where
Sorting
Aggregation
Grouping
Having

Tutorial Exercises

Conclusion Suggested Readings

GROUP BY columnList

HAVING condition

- HAVING clause is designed for use with GROUP BY to restrict groups that appear in final result table
- Similar to WHERE, but WHERE filters individual rows whereas HAVING filters groups
- Column names in HAVING clause must also appear in the GROUP BY list or be contained within an aggregate function

HAVING: Example

Data Retrieval

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000.00	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000.00	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000.00	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000.00	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000.00	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000.00	B005

```
SELECT branchNo,  
COUNT(staffNo) AS count,  
SUM(salary) AS sum FROM Staff  
GROUP BY branchNo HAVING COUNT(staffNo) > 1
```

- What does this query?

Exercise 1

Data Retrieval

Introduction

Retrieval
Select
Where
Sorting
Aggregation
Grouping
Having

Tutorial Exercises

Conclusion

Suggested
Readings

Branch (branchNo, street, city, postcode)
Staff (staffNo, fName, lName, position, sex, DOB, salary, branchNo)
PropertyForRent (propertyNo, street, city, postcode, type, rooms,
rent, ownerNo, staffNo, branchNo)

Produce a list of monthly salaries for all staff, showing only the staff number, the first and last names, and the salary details.

Exercise 2

Data Retrieval

Introduction

Retrieval
Select
Where
Sorting
Aggregation
Grouping
Having

Tutorial Exercises

Conclusion
Suggested
Readings

Branch (branchNo, street, city, postcode)
Staff (staffNo, fName, lName, position, sex, DOB, salary, branchNo)
PropertyForRent (propertyNo, street, city, postcode, type, rooms,
rent, ownerNo, staffNo, branchNo)

List the names of all cities where there is a branch

Exercise 3

Data Retrieval

Introduction

Retrieval
Select
Where
Sorting
Aggregation
Grouping
Having

Tutorial Exercises

Conclusion

Suggested
Readings

Branch (branchNo, street, city, postcode)
Staff (staffNo, fName, lName, position, sex, DOB, salary, branchNo)
PropertyForRent (propertyNo, street, city, postcode, type, rooms,
rent, ownerNo, staffNo, branchNo)

List all properties whose that are located in 'London' or
'Liverpool' in alphabetical order of street.

Exercise 4

Data Retrieval

Introduction

Retrieval
Select
Where
Sorting
Aggregation
Grouping
Having

Tutorial Exercises

Conclusion

Suggested
Readings

Branch (branchNo, street, city, postcode)
Staff (staffNo, fName, lName, position, sex, DOB, salary, branchNo)
PropertyForRent (propertyNo, street, city, postcode, type, rooms,
rent, ownerNo, staffNo, branchNo)

How many different members of the staff are managing properties that cost more than 10000 and aren't in London

Exercise 5

Data Retrieval

Introduction

Retrieval
Select
Where
Sorting
Aggregation
Grouping
Having

Tutorial Exercises

Conclusion
Suggested
Readings

Branch (branchNo, street, city, postcode)
Staff (staffNo, fName, lName, position, sex, DOB, salary, branchNo)
PropertyForRent (propertyNo, street, city, postcode, type, rooms,
rent, ownerNo, staffNo, branchNo)

What is the average salary managers in each branch?

Exercise 6

Data Retrieval

Introduction

Retrieval
Select
Where
Sorting
Aggregation
Grouping
Having

Tutorial Exercises

Conclusion

Suggested
Readings

Branch (branchNo, street, city, postcode)
Staff (staffNo, fName, lName, position, sex, DOB, salary, branchNo)
PropertyForRent (propertyNo, street, city, postcode, type, rooms,
rent, ownerNo, staffNo, branchNo)

Display the id of all members of the staff that manage more than 10 properties

Exercise 7

Data Retrieval

Introduction

Retrieval
Select
Where
Sorting
Aggregation
Grouping
Having

Tutorial Exercises

Conclusion

Suggested
Readings

Branch (branchNo, street, city, postcode)
Staff (staffNo, fName, lName, position, sex, DOB, salary, branchNo)
PropertyForRent (propertyNo, street, city, postcode, type, rooms,
rent, ownerNo, staffNo, branchNo)

Display the total cost of all flats managed for each member of the staff and each branch

Exercise 8

Data Retrieval

Introduction

Retrieval
Select
Where
Sorting
Aggregation
Grouping
Having

Tutorial Exercises

Conclusion Suggested Readings

Branch (branchNo, street, city, postcode)
Staff (staffNo, fName, lName, position, sex, DOB, salary, branchNo)
PropertyForRent (propertyNo, street, city, postcode, type, rooms,
rent, ownerNo, staffNo, branchNo)

Display the id of all owners that have flats managed by different branches

Conclusion

Data Retrieval

Introduction

Retrieval
Select
Where
Sorting
Aggregation
Grouping
Having

Tutorial Exercises

Conclusion

Suggested
Readings

In this session we have covered:

- Introduction to SQL
- Select Statement:
 - Select
 - Where
 - Order by
 - Aggregation
 - Group by
 - Having

Lab Session

Data Retrieval

Introduction

Retrieval

Select

Where

Sorting

Aggregation

Grouping

Having

Tutorial

Exercises

Conclusion

Suggested
Readings

This session is about performing simple SELECT queries in
MySQL

Suggested Readings

Data Retrieval

Introduction

Retrieval

Select

Where

Sorting

Aggregation

Grouping

Having

Tutorial

Exercises

Conclusion

Suggested Readings

- Chapter 4 of Fundamentals of Database Systems. Elmasri & Navathe.
- Chapter 5 of Database systems: a practical approach to design, implementation, and management. Connolly, Thomas M; Begg, Carolyn