Course Introduction

Welcome

Mostafa
Nabieh

Mostafa
Nabieh

EPITA
ÉCOLE D'INGÉNIEURS EN INFORMATIQUE

HUAWEI

IBM

AI TALENTS HUB

Why SQL for Data Science

Why SQL for Data Science

- **Median based salary: $110,000**

Why SQL for Data Science

- Median based salary: $110,000
- Job satisfaction score: 4.4/5

Why SQL for Data Science

- Median based salary: $110,000
- Job satisfaction score: 4.4/5
- Top spot on Glassoor's best jobs in America

Why SQL for Data Science

- Median based salary: $110,000
- Job satisfaction score: 4.4/5
- Top spot on Glassoor's best jobs in America
- Top three skills for a Data Scientist

Why SQL for Data Science

Why SQL for Data Science

- Big data

Why SQL for Data Science

- Big data
- Table with a few rows
- Script

Why SQL for Data Science

- Big data
- Table with a few rows
- Small start up
- Big Database

Why SQL for Data Science

- Big data
- Table with a few rows
- Small start up
- Big Database
- Mobile phone

Why SQL for Data Science

Advantages:

Why SQL for Data Science

Advantages:

- Boost your professional profile

Why SQL for Data Science

Advantages:

- Boost your professional profile
- Give you a good understanding of relational databases

Why SQL for Data Science

Course details

- Basics for SQL and Relational Databases

Course details

- Basics for SQL and Relational Databases
- Working knowledge of SQL and Databases

Course details

- Basics for SQL and Relational Databases
- Working knowledge of SQL and Databases
- Connect to Database and run SQL queries
- Python and Jupyter Notebooks to Analyze Data

Course details

- Basics for SQL and Relational Databases
- Working knowledge of SQL and Databases
- Connect to Database and run SQL queries
- Python and Jupyter Notebooks to Analyze Data
- Assignment to apply concepts with Real World Dataset

Introduction to Databases

Course Overview

- Basics of SQL
- Relational Database Model

- At the end of this course, you will be able to discuss SQL basics and explain aspects of the relational database model

- At the end of this lesson, you will be able to:
  - Describe SQL, data, database, relational database
  - List five basic SQL commands

What is SQL?

- A language used for relational databases
- Query data

What is data?

- Facts (words, numbers)

What is data?

- Facts (words, numbers)
- Pictures
- One of the most critical assets of any business
- Needs to be secure

What is a database?

- A repository of data
- Provides the functionality for adding, modifying and querying that data

What is a database?

- A repository of data
- Provides the functionality for adding, modifying and querying that data
- Different kinds of databases store data in different forms

Relational Database

- Data stored in tabular form - columns and rows

Student ID	First Name	Last Name
34933	Victoria	Slater
93759	Justin	McNeil
20847	Jessica	Bennett
65947	Michelle	Dolin
24956	David	Price
65692	Franklin	Mullins
24271	Alissa	Lee

Relational Database

- Data stored in tabular form - columns and rows
- Columns contain item properties e.g. Last Name, First Name, etc.

Student ID	First Name	Last Name
34933	Victoria	Slater
93759	Justin	McNeil
20847	Jessica	Bennett
65947	Michelle	Dolin
24956	David	Price
65692	Franklin	Mullins
24271	Alissa	Lee

Relational Database

- Data stored in tabular form - columns and rows
- Columns contain item properties e.g. Last Name, First Name, etc.
- Table is collection of related things e.g. Employees, Authors, etc.

Student ID	First Name	Last Name
34933	Victoria	Slater
93759	Justin	McNeil
20847	Jessica	Bennett
65947	Michelle	Dolin
24956	David	Price
65692	Franklin	Mullins
24271	Alissa	Lee

Relational Database

- Data stored in tabular form - columns and rows
- Columns contain item properties e.g. Last Name, First Name, etc.
- Table is collection of related things e.g. Employees, Authors, etc.
- Relationships can exist between tables

  (hence: "relational")

Student ID	First Name	Last Name
34933	Victoria	Slater
93759	Justin	McNeil
20847	Jessica	Bennett
65947	Michelle	Dolin
24956	David	Price
65692	Franklin	Mullins
24271	Alissa	Lee

DBMS

- Database: repository of data

DBMS

- Database: repository of data
- DBMS: Database Management System - software to manage databases

DBMS

- Database: repository of data
- DBMS: Database Management System - software to manage databases
- Database, Database Server, Database System, Data Server, DBMS - often used interchangeably

What is RDBMS?

- RDBMS = Relational database management system

What is RDBMS?

- RDBMS = Relational database management system
- A set of software tools that controls the data

What is RDBMS?

- RDBMS = Relational database management system
- A set of software tools that controls the data
  - access, organization, and storage

What is RDBMS?

- RDBMS = Relational database management system
- A set of software tools that controls the data
  - access, organization, and storage
- Examples are: MySQL, Oracle Database, IBM Db2, etc.

Basic SQL Commands

- Create a table

Basic SQL Commands

- Create a table
- Insert

Basic SQL Commands

- Create a table
- Insert
- Select
- Update

Basic SQL Commands

- Create a table
- Insert
- Select
- Update
- Delete

Summary

- You can now describe:
  - SQL
  - Data
  - Database
  - Relational Databases
  - RDBMS
  - 5 basic SQL commands:
    - Create, Insert, Select, Update, Delete

SELECT statement

Retrieving data from a table

Retrieving rows from a table

- After creating a table and inserting data into the table, we want to see the data

Retrieving rows from a table

- After creating a table and inserting data into the table, we want to see the data
- SELECT statement

Retrieving rows from a table

- After creating a table and inserting data into the table, we want to see the data
- SELECT statement
  - A Data Manipulation Language (DML) statement used to read and modify data

Retrieving rows from a table

- After creating a table and inserting data into the table, we want to see the data
- SELECT statement
  - A Data Manipulation Language (DML) statement used to read and modify data

```
Select statement:    Query
Result from the query:  Result set/table

Select * from <tablename>
```

Using the SELECT Statement

Title	Edition	Year	Price	ISBN	Pages	Aisle	Description
Database Fundamentals	1	2010	24.99	978-0-9800628 3-1-1	300	DB-A02	Teaches you the fundamentals of databases
Getting started with DB2 Express-C	1	2010	24.99	978-0-9866628 3-5-1	280	DB-A01	Teaches you the essentials of DB2 using DB2 Express-C

Using the SELECT Statement

Title	Edition	Year	Price	ISBN	Pages	Aisle	Description
Database Fundamentals	1	2010	24.99	978-0-9800628 3-1-1	300	DB-A02	Teaches you the fundamentals of databases
Getting started with DB2 Express-C	1	2010	24.99	978-0-9866628 3-5-1	280	DB-A01	Teaches you the essentials of DB2 using DB2 Express-C

Example: select * from Book

```
db2 => select * from Book
```

Using the SELECT Statement

Title	Edition	Year	Price	ISBN	Pages	Aisle	Description
Database Fundamentals	1	2010	24.99	978-0-9800628 3-1-1	300	DB-A02	Teaches you the fundamentals of databases
Getting started with DB2 Express-C	1	2010	24.99	978-0-9866628 3-5-1	280	DB-A01	Teaches you the essentials of DB2 using DB2 Express-C

Example: select * from Book

```
db2 => select * from Book
```

Book_ID	Title	Edition	Year	Price	ISBN	Pages	Aisle	Description
B1	Getting started with DB2 Express-C	1	2010	24.99	978-0-98666283-5-1	280	DB-A01	Teaches you the essentials of DB2 using DB2 Express-C
B2	Database Fundamentals	1	2010	24.99	978-0-98006283-1-1	300	DB-A02	Teaches you the fundamentals of databases
B3	Getting started with DB2 App Dev	1	2011	35.99	978-0-98086283-4-1	345	DB-A03	Teaches you the essentials of developing applications for DB2.
B4	Getting started with WAS CE	1	2010	49.99	978-0-98946283-3-1	458	DB-A04	Teaches you the essentials of WebSphere Application Server

```
4 record(s) selected.
```

Using the SELECT Statement

Title	Edition	Year	Price	ISBN	Pages	Aisle	Description
Database Fundamentals	1	2010	24.99	978-0-9800628 3-1-1	300	DB-A02	Teaches you the fundamentals of databases
Getting started with DB2 Express-C	1	2010	24.99	978-0-9866628 3-5-1	280	DB-A01	Teaches you the essentials of DB2 using DB2 Express-C

Example: select <column 1, column 2, ..., column n from Book

```
db2 => select book_id, title, edition, year, price, ISBN, pages, aisle, description from Book
```

Book_ID	Title	Edition	Year	Price	ISBN	Pages	Aisle	Description
B1	Getting started with DB2 Express-C	1	2010	24.99	978-0-98666283-5-1	280	DB-A01	Teaches you the essentials of DB2 using DB2 Express-C
B2	Database Fundamentals	1	2010	24.99	978-0-98006283-1-1	300	DB-A02	Teaches you the fundamentals of databases
B3	Getting started with DB2 App Dev	1	2011	35.99	978-0-98086283-4-1	345	DB-A03	Teaches you the essentials of developing applications for DB2.
B4	Getting started with WAS CE	1	2010	49.99	978-0-98946283-3-1	458	DB-A04	Teaches you the essentials of WebSphere Application Server

```
4 record(s) selected.
```

Retrieving a subset of the columns

Retrieving a subset of the columns

- You can retrieve just the columns you want

Retrieving a subset of the columns

- You can retrieve just the columns you want
- The order of the columns displayed always matches the order in the SELECT statement

Retrieving a subset of the columns

- You can retrieve just the columns you want
- The order of the columns displayed always matches the order in the SELECT statement
  - `SELECT <column 1>, <column 2> from Book`

Retrieving a subset of the columns

- You can retrieve just the columns you want
- The order of the columns displayed always matches the order in the SELECT statement
  - `SELECT <column 1>, <column 2> from Book`

```
db2 => select book_id, title from Book

Book_ID     Title
----------------------------------------
B1          Getting started with DB2 Express-C
B2          Database Fundamentals
B3          Getting started with DB2 App Dev
B4          Getting started with WAS CE

     4 record(s) selected.
```

Restricting the Result Set: WHERE Clause

- Restricts the result set
- Always requires a Predicate:
  - Evaluates to:
    True, False or Unknown

```
select book_id, title from Book
       WHERE predicate
```

Restricting the Result Set: WHERE Clause

- Restricts the result set
- Always requires a Predicate:
  - Evaluates to:
    True, False or Unknown
  - Used in the search condition of the Where clause

```
select book_id, title from Book
       WHERE predicate
```

Restricting the Result Set: WHERE Clause

- Restricts the result set
- Always requires a Predicate:
  - Evaluates to:
    True, False or Unknown
  - Used in the search condition of the Where clause

```
select book_id, title from Book
       WHERE predicate
```

```
db2 => select book_id, title from Book
       WHERE book_id='B1'
```

Restricting the Result Set: WHERE Clause

- Restricts the result set
- Always requires a Predicate:
  - Evaluates to:
    True, False or Unknown
  - Used in the search condition of the Where clause

```
select book_id, title from Book
       WHERE predicate
```

```
db2 => select book_id, title from Book
       WHERE book_id='B1'

Book_ID    Title
-----------------------------------------------------------
B1         Getting started with DB2 Express-C

  1 record(s) selected
```

WHERE Clause Comparison Operators

```
select book_id, title from Book
    WHERE book_id = 'B1'
```

Equal to	=
Greater than	>
Lesser than	<
Greater than or equal to	>=
Less than or equal to	<=
Not equal to	<>

Summary

Now you can:

- Retrieve data from a relational database table
- Define the use of a predicate
- Identify the syntax of the SELECT statement using the WHERE clause
- List the comparison operators supported by a RDBMS

COUNT

COUNT() - a built-in function that retrieves the number of rows matching the query criteria.

COUNT

COUNT() - a built-in function that retrieves the number of rows matching the query criteria.

Number of rows in a table:

```
select COUNT(*) from tablename
```

COUNT

Rows in the MEDALS table where Country is Canada:

```
select COUNT(COUNTRY) from MEDALS
        where COUNTRY='CANADA'
```

Result:

```
                1
---------------
               29
```

DISTINCT

DISTINCT is used to remove duplicate values from a result set.

Retrieve unique values in a column:

```
select DISTINCT columnname from tablename
```

DISTINCT

List of unique countries that received GOLD medals:

```
select DISTINCT COUNTRY from MEDALS
        where MEDALTYPE = 'GOLD'
```

DISTINCT

List of unique countries that received GOLD medals:

```
select DISTINCT COUNTRY from MEDALS
        where MEDALTYPE = 'GOLD'
```

Result:

```
                1
---------------
               21
```

LIMIT

LIMIT is used for restricting the number of rows retrieved from the database.

Retrieve just the first 10 rows in a table:

```
select * from tablename LIMIT 10
```

LIMIT

LIMIT is used for restricting the number of rows retrieved from the database.

Retrieve just the first 10 rows in a table:

```
select * from tablename LIMIT 10
```

LIMIT

Retrieve 5 rows in the MEDALS table for a particular year:

```
select * from MEDALS
        where YEAR = 2018 LIMIT 5
```

Result:

COUNTRY	GOLD	SILVER	BRONZE	TOTAL	YEAR
Norway	14	14	11	39	2018
Germany	14	10	7	31	2018
Canada	11	8	10	29	2018
United States	9	8	6	23	2018
Netherlands	8	6	6	20	2018

INSERT Statement

Adding rows to a table

- Create the table (CREATE TABLE statement)
- Populate table with data:
  - INSERT statement
    - a Data Manipulation Language (DML) statement used to read and modify data

Author_ID	Lastname	Firstname	Email	City	Country
A1	Chong	Raul	rfc@ibm.com	Toronto	CA
A2	Ahuja	Rav	ra@ibm.com	Toronto	CA
A3	Hakes	Ian	ih@ibm.com	Toronto	Ca
A4	Sharma	Neeraj	ns@ibm.com	Chennai	IN
A5	Perniu	Liviu	lp@ibm.com	Transylvania	RO

Using the INSERT Statement

Author_ID	Lastname	Firstname	Email	City	Country
A1	Chong	Raul	rfc@ibm.com	Toronto	CA
A2	Ahuja	Rav	ra@ibm.com	Toronto	CA
A3	Hakes	Ian	ih@ibm.com	Toronto	Ca
A4	Sharma	Neeraj	ns@ibm.com	Chennai	IN
A5	Perniu	Liviu	lp@ibm.com	Transylvania	RO

Using the INSERT Statement

Author_ID	Lastname	Firstname	Email	City	Country
A1	Chong	Raul	rfc@ibm.com	Toronto	CA
A2	Ahuja	Rav	ra@ibm.com	Toronto	CA
A3	Hakes	Ian	ih@ibm.com	Toronto	Ca
A4	Sharma	Neeraj	ns@ibm.com	Chennai	IN
A5	Perniu	Liviu	lp@ibm.com	Transylvania	RO

```
INSERT INTO [TableName]
    <([ColumnName],...)>
VALUES ([Value],...)
```

Using the INSERT Statement

Author_ID	Lastname	Firstname	Email	City	Country
A1	Chong	Raul	rfc@ibm.com	Toronto	CA
A2	Ahuja	Rav	ra@ibm.com	Toronto	CA
A3	Hakes	Ian	ih@ibm.com	Toronto	Ca
A4	Sharma	Neeraj	ns@ibm.com	Chennai	IN
A5	Perniu	Liviu	lp@ibm.com	Transylvania	RO

```
INSERT INTO [TableName]
    <([ColumnName],...)>
VALUES ([Value],...)
```

```
INSERT INTO AUTHOR
    (AUTHOR_ID, LASTNAME, FIRSTNAME, EMAIL, CITY, COUNTRY)
VALUES ('A1', 'Chong', 'Raul', 'rfc@ibm.com', 'Toronto', 'CA')
```

Inserting multiple rows

Author_ID	Lastname	Firstname	Email	City	Country
A1	Chong	Raul	rfc@ibm.com	Toronto	CA
A2	Ahuja	Rav	ra@ibm.com	Toronto	CA
A3	Hakes	Ian	ih@ibm.com	Toronto	Ca
A4	Sharma	Neeraj	ns@ibm.com	Chennai	IN
A5	Perniu	Liviu	lp@ibm.com	Transylvania	RO

```
INSERT INTO AUTHOR
    (AUTHOR_ID, LASTNAME, FIRSTNAME, EMAIL, CITY, COUNTRY)
VALUES
    ('A1', 'Chong', 'Raul', 'rfc@ibm.com', 'Toronto', 'CA')
    ('A2', 'Ahuja', 'Rav', 'ra@ibm.com', 'Toronto', 'CA')
```

Summary

Now you can:

- Identify the syntax of the INSERT statement
- Explain two methods to add rows to a table

UPDATE and DELETE Statements

Altering rows of a table – UPDATE statement

- After creating a table and inserting data into the table, we can alter the data

Altering rows of a table – UPDATE statement

- After creating a table and inserting data into the table, we can alter the data
  - UPDATE statement: A Data Manipulation Language (DML) statement used to read and modify data

Author_Id	LastName	FirstName	Email	City	Country
A1	Chong	Raul	rfc@ibm.com	Toronto	CA
A2	Ahuja	Rav	ra@ibm.com	Toronto	CA
A3	Hakes	Ian	ih@ibm.com	Toronto	CA
A4	Sharma	Neeraj	ns@ibm.com	Chennai	IN
A5	Perniu	Liviu	lp@ibm.com	Transylvania	RO

Using the UPDATE Statement

Author_Id	LastName	FirstName	Email	City	Country
A1	Chong	Raul	rfc@ibm.com	Toronto	CA
A2	Ahuja	Rav	ra@ibm.com	Toronto	CA
A3	Hakes	Ian	ih@ibm.com	Toronto	CA
A4	Sharma	Neeraj	ns@ibm.com	Chennai	IN
A5	Perniu	Liviu	lp@ibm.com	Transylvania	RO

Using the UPDATE Statement

Author_Id	LastName	FirstName	Email	City	Country
A1	Chong	Raul	rfc@ibm.com	Toronto	CA
A2	Ahuja	Rav	ra@ibm.com	Toronto	CA
A3	Hakes	Ian	ih@ibm.com	Toronto	CA
A4	Sharma	Neeraj	ns@ibm.com	Chennai	IN
A5	Perniu	Liviu	lp@ibm.com	Transylvania	RO

UPDATE [TableName]
SET [[ColumnName]=[Value]]
<WHERE [Condition]>

Using the UPDATE Statement

Author_Id	LastName	FirstName	Email	City	Country
A1	CHONG	RAUL	rfc@ibm.com	Toronto	CA
A2	AHUJA	RAV	ra@ibm.com	Toronto	CA
A3	HAKES	IAN	ih@ibm.com	Toronto	CA

```
UPDATE AUTHOR
SET LASTNAME='KATTA'
    FIRSTNAME='LAKSHMI'
    WHERE AUTHOR_ID='A2'
```

Using the UPDATE Statement

Author_Id	LastName	FirstName	Email	City	Country
A1	CHONG	RAUL	rfc@ibm.com	Toronto	CA
A2	AHUJA	RAV	ra@ibm.com	Toronto	CA
A3	HAKES	IAN	ih@ibm.com	Toronto	CA

```
UPDATE AUTHOR
SET LASTNAME='KATTA'
    FIRSTNAME='LAKSHMI'
    WHERE AUTHOR_ID='A2'
```

Author_Id	LastName	FirstName	Email	City	Country
A1	CHONG	RAUL	rfc@ibm.com	Toronto	CA
A2	KATTA	LAKSHMI	ra@ibm.com	Toronto	CA
A3	HAKES	IAN	ih@ibm.com	Toronto	CA

Deleting Rows from a table

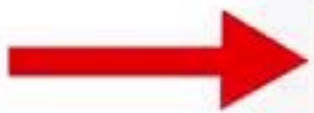
- Remove 1 or more rows from the table:
  - DELETE statement
    - A DML statement used to read and modify data

DELETE FROM [TableName]
<WHERE [Condition]>

Using the DELETE Statement

Author_Id	LastName	FirstName	Email	City	Country
A1	Chong	Raul	rfc@ibm.com	Toronto	CA
A2	Ahuja	Rav	ra@ibm.com	Toronto	CA
A3	Hakes	Ian	ih@ibm.com	Toronto	CA
A4	Sharma	Neeraj	ns@ibm.com	Chennai	IN
A5	Perniu	Liviu	lp@ibm.com	Transylvania	RO

```
DELETE FROM AUTHOR
    WHERE AUTHOR_ID IN ('A2', 'A3')
```



Author_Id	LastName	FirstName	Email	City	Country
A1	Chong	Raul	rfc@ibm.com	Toronto	CA
A4	Sharma	Neeraj	ns@ibm.com	Chennai	IN
A5	Perniu	Liviu	lp@ibm.com	Transylvania	RO

Summary

Now you can:

- Identify the syntax of the UPDATE statement
- Identify the syntax of the DELETE statement
- Explain the importance of the WHERE clause in both the UPDATE and DELETE statements