DATABASES

Data and Databases

- What is data?
  - Facts (words, numbers), pictures…

- What is a database?
  - A repository of data
  - Provides the functionality for adding, modifying and querying that data

Crystallographic databases

PDB: Protein Data Bank

NDB: Nucleic Acid Database

COD: Crystallography Open Database

CSD: Cambridge Structural Database

ICSD: Inorganic Crystal Structure Database

PDF: Powder Diffraction File of the International Centre for Diffraction Data

BCS: Bilbao Crystallographic Server

Practical examples of crystallographic db

- Search for **homologous proteins**
- **AlphaFold** to predict the 3D structure based on its sequence:
  - Trained on data consisting of ~170,000 protein structures from the PDB together with large databases containing 180 million protein sequences
- **Model building:** *eg.* generation of side chains using rotamer library
- **Refinement:** geometric and stereochemical restraints
- **Validation:** Ramachandran plot derived from an analysis of a set of well refined and high-resolution crystal structures

Relational Databases

- Different kinds of databases store data in different forms: the most popular is called relational database

- Data stored in tables – columns and rows
  - A row represents an item
  - Columns contain item properties

PDB_id	Resolution	Residues
3AZD	0.98	74
2FXM	2.70	258
3MQB	3.20	484

- A relational database can contain multiple, related tables

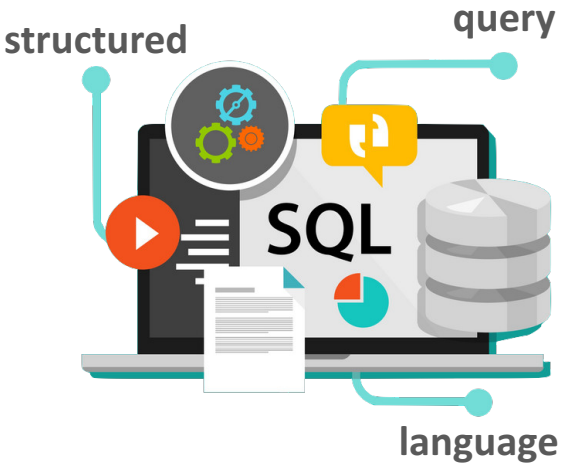
DBMS: DataBase Management System

- Database Management system (DBMS) – software to manage databases
- Relational database management system (RDBMS) – software to manage relational databases
- A set of software tools that controls the data:
  - access, organization, storage and visualization
- Examples include:

- They uses SQL as the standard query language

What is SQL?

- SQL stands for **S**tructured **Q**uery **L**anguage
- Original name SEQUEL
  (**S**tructured **E**nglish **Que**ry **L**anguage)
- SQL is a language used to communicate with relational databases
- The most widely implemented database language



Types of SQL statements

- SQL statements are used for interacting with tables, columns and rows

SQL STATEMENT TYPES:

- **DDL** (**D**ata **D**efinition **L**anguage):
  - Define, change, or drop data
- Common DDL:
  - CREATE
  - ALTER
  - TRUNCATE
  - DROP

- **DML** (**D**ata **M**anipulation **L**anguage):
  - Read and modify data
- Common DML:
  - INSERT
  - SELECT
  - UPDATE
  - DELETE

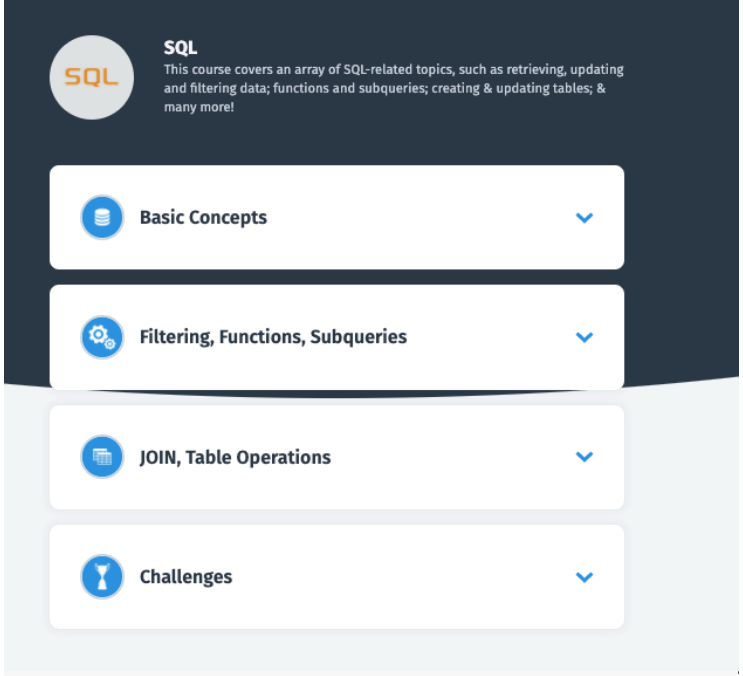
Where to learn SQL?

SOLOLEARN

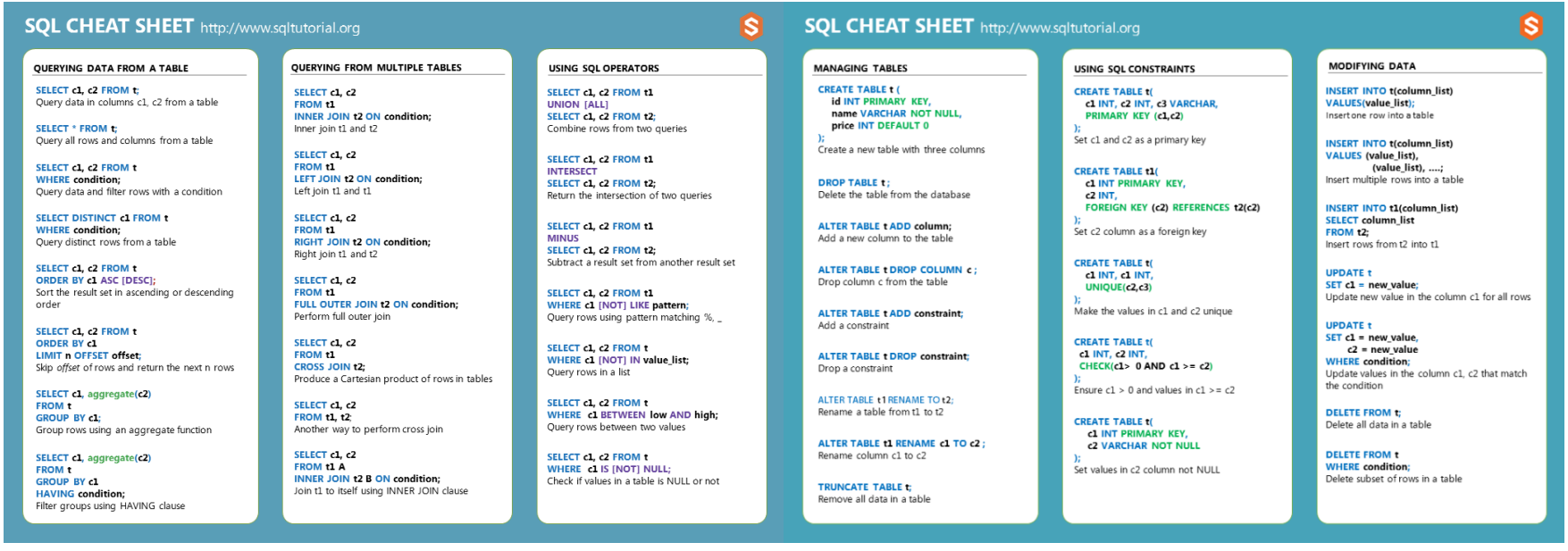
SQL

SQL
Fundamentals
159 XP

OPEN COURSE | LEADERBOARD



SQL Cheat Sheets



SQL CHEAT SHEET http://www.sqltutorial.org

QUERYING DATA FROM A TABLE

```
SELECT c1, c2 FROM t;
```
Query data in columns c1, c2 from a table

```
SELECT * FROM t;
```
Query all rows and columns from a table

```
SELECT c1, c2 FROM t
WHERE condition;
```
Query data and filter rows with a condition

```
SELECT DISTINCT c1 FROM t
WHERE condition;
```
Query distinct rows from a table

```
SELECT c1, c2 FROM t
ORDER BY c1 ASC [DESC];
```
Sort the result set in ascending or descending order

```
SELECT c1, c2 FROM t
ORDER BY c1
LIMIT n OFFSET offset;
```
Skip *offset* of rows and return the next n rows

```
SELECT c1, aggregate(c2)
FROM t
GROUP BY c1;
```
Group rows using an aggregate function

```
SELECT c1, aggregate(c2)
FROM t
GROUP BY c1
HAVING condition;
```
Filter groups using HAVING clause

QUERYING FROM MULTIPLE TABLES

```
SELECT c1, c2
FROM t1
INNER JOIN t2 ON condition;
```
Inner join t1 and t2

```
SELECT c1, c2
FROM t1
LEFT JOIN t2 ON condition;
```
Left join t1 and t1

```
SELECT c1, c2
FROM t1
RIGHT JOIN t2 ON condition;
```
Right join t1 and t2

```
SELECT c1, c2
FROM t1
FULL OUTER JOIN t2 ON condition;
```
Perform full outer join

```
SELECT c1, c2
FROM t1
CROSS JOIN t2;
```
Produce a Cartesian product of rows in tables

```
SELECT c1, c2
FROM t1, t2;
```
Another way to perform cross join

```
SELECT c1, c2
FROM t1 A
INNER JOIN t2 B ON condition;
```
Join t1 to itself using INNER JOIN clause

USING SQL OPERATORS

```
SELECT c1, c2 FROM t1
UNION [ALL]
SELECT c1, c2 FROM t2;
```
Combine rows from two queries

```
SELECT c1, c2 FROM t1
INTERSECT
SELECT c1, c2 FROM t2;
```
Return the intersection of two queries

```
SELECT c1, c2 FROM t1
MINUS
SELECT c1, c2 FROM t2;
```
Subtract a result set from another result set

```
SELECT c1, c2 FROM t1
WHERE c1 [NOT] LIKE pattern;
```
Query rows using pattern matching %, _

```
SELECT c1, c2 FROM t
WHERE c1 [NOT] IN value_list;
```
Query rows in a list

```
SELECT c1, c2 FROM t
WHERE c1 BETWEEN low AND high;
```
Query rows between two values

```
SELECT c1, c2 FROM t
WHERE c1 IS [NOT] NULL;
```
Check if values in a table is NULL or not

SQL CHEAT SHEET http://www.sqltutorial.org

MANAGING TABLES

```
CREATE TABLE t (
    id INT PRIMARY KEY,
    name VARCHAR NOT NULL,
    price INT DEFAULT 0
);
```
Create a new table with three columns

```
DROP TABLE t;
```
Delete the table from the database

```
ALTER TABLE t ADD column;
```
Add a new column to the table

```
ALTER TABLE t DROP COLUMN c;
```
Drop column c from the table

```
ALTER TABLE t ADD constraint;
```
Add a constraint

```
ALTER TABLE t DROP constraint;
```
Drop a constraint

```
ALTER TABLE t1 RENAME TO t2;
```
Rename a table from t1 to t2

```
ALTER TABLE t1 RENAME c1 TO c2;
```
Rename column c1 to c2

```
TRUNCATE TABLE t;
```
Remove all data in a table

USING SQL CONSTRAINTS

```
CREATE TABLE t(
   c1 INT, c2 INT, c3 VARCHAR,
   PRIMARY KEY (c1,c2)
);
```
Set c1 and c2 as a primary key

```
CREATE TABLE t1(
   c1 INT PRIMARY KEY,
   c2 INT,
   FOREIGN KEY (c2) REFERENCES t2(c2)
);
```
Set c2 column as a foreign key

```
CREATE TABLE t(
   c1 INT, c1 INT,
   UNIQUE(c2,c3)
);
```
Make the values in c1 and c2 unique

```
CREATE TABLE t(
  c1 INT, c2 INT,
  CHECK(c1> 0 AND c1 >= c2)
);
```
Ensure c1 > 0 and values in c1 >= c2

```
CREATE TABLE t(
   c1 INT PRIMARY KEY,
   c2 VARCHAR NOT NULL
);
```
Set values in c2 column not NULL

MODIFYING DATA

```
INSERT INTO t(column_list)
VALUES(value_list);
```
Insert one row into a table

```
INSERT INTO t(column_list)
VALUES (value_list),
       (value_list), ....;
```
Insert multiple rows into a table

```
INSERT INTO t1(column_list)
SELECT column_list
FROM t2;
```
Insert rows from t2 into t1

```
UPDATE t
SET c1 = new_value;
```
Update new value in the column c1 for all rows

```
UPDATE t
SET c1 = new_value,
    c2 = new_value
WHERE condition;
```
Update values in the column c1, c2 that match the condition

```
DELETE FROM t;
```
Delete all data in a table

```
DELETE FROM t
WHERE condition;
```
Delete subset of rows in a table

- https://www.sqltutorial.org/wp-content/uploads/2016/04/SQL-cheat-sheet.pdf
- https://learnsql.com/blog/sql-basics-cheat-sheet/sql-basics-cheat-sheet-a4.pdf
- https://intellipaat.com/mediaFiles/2019/02/SQL-Commands-Cheat-Sheet.pdf