

1. *What is the difference between a “Data Warehouse” and a “Federated Database”?*

- **Data Warehouse:** Is a central repository for all or significant parts of the data that an enterprise's various business systems collect. Typically, a data warehouse is housed on an enterprise mainframe server or increasingly, in the cloud.
- **Federated Database:** Is a database management system, which transparently maps multiple autonomous database systems into a single federated database. This means that multiple databases appears as one database to the user.

2. *Name and describe at least four typical advantages of DBMS - based data storage over file system based data management. Give one real world example for each.*

- (a) A locking mechanism for concurrent access (If users access the same item in the database there is no RW or WR errors.)
- (b) The ability to swiftly recover from crashes and errors, including restartability and recoverability (If the System Crashes due to for example power failures the Logging system recovers know changes - hence reducing loss)
- (c) Robust data integrity capabilities (Data is the same from all access points).
- (d) Logging and auditing of activity (If changes are discarded before saving the system can re-roll.)

3. Design and paint an Entity Relationship (ER) diagram for a small database dedicated to store gene regulatory interactions of a bacterium. The following entities are necessary: genes (with ID and one or more (!) names), transcription factors (with ID and one name) and binding sites (with binding sequence of 10...30 bp length). Include the following relations:

- (a) A transcription factor is encoded by one gene.
- (b) A transcription factor may regulate a gene by binging to one or more binding sites.

Which approach?

- **Ontological:** If I would need a quick overview, where I don't know if there is any structure in the data or what the structure is.
- **Standard:** Once a structure of the data has revealed itself.

A SQL code

```
1
2  -- Definitions
3  DROP DATABASE IF EXISTS bioInfo;
4  CREATE DATABASE bioInfo; -- create new database
5  USE bioInfo; -- use that new database
6
7
8  CREATE TABLE GENE (
9      g_sequence CHAR(32) NOT NULL,
10     id INT PRIMARY KEY AUTO_INCREMENT
11 );
12
13 CREATE TABLE TRANSCRIPTION_FACTOR (
14     tf_name CHAR(32) NOT NULL,
15     id INT PRIMARY KEY AUTO_INCREMENT,
16     FOREIGN KEY (b_id) REFERENCES (BINDING.id)
17 );
18
19 CREATE TABLE BINDING (
20     id INT PRIMARY KEY AUTO_INCREMENT,
21     b_sequence CHAR(32) NOT NULL,
22     FOREIGN KEY (g_id) REFERENCES (GENE.id)
23 );
24
25 CREATE VIEW regulation AS
26     SELECT g_id AS id, tf_name, b_sequence, g_sequence
27     FROM TRANSCRIPTION_FACTOR AS A JOIN BINDING AS B JOIN GENE AS C
28     ON A.b_id = B.id AND C.id = B.g_id;
29
30 CREATE VIEW is_called AS
31     SELECT g_id AS id, g_name AS gene_name, g_sequence AS gene_sequence
32     FROM GENE AS A JOIN GENE_NAME AS B
33     ON A.id = B.g_id;
34
35 CREATE VIEW encodes AS
36     SELECT b_id AS id, g_sequence AS gene_sequence, tf_name as transcription_factor
37     FROM GENE AS A JOIN TRANSCRIPTION_FACTOR AS B
38     ON A.id = B.b_id;
39
40 -- Filler data
41 INSERT INTO GENE (g_sequence) VALUES ("ATTAT");
42 INSERT INTO GENE (g_sequence) VALUES ("GTDAT");
43 INSERT INTO GENE (g_sequence) VALUES ("GGADT");
44
45 INSERT INTO GENE_NAME (g_name, g_id) VALUES ("kl0022", 1);
46 INSERT INTO GENE_NAME (g_name, g_id) VALUES ("1072er", 2);
47 INSERT INTO GENE_NAME (g_name, g_id) VALUES ("xx06a70", 3);
48
49 INSERT INTO TRANSCRIPTION_FACTOR (tf_name, b_id) VALUES ("protein0", 1);
50 INSERT INTO TRANSCRIPTION_FACTOR (tf_name, b_id) VALUES ("protein1", 2);
51 INSERT INTO TRANSCRIPTION_FACTOR (tf_name, b_id) VALUES ("protein2", 3);
52
53 INSERT INTO BINDING (b_sequence, g_id) VALUES ("AATTTAT", 1);
54 INSERT INTO BINDING (b_sequence, g_id) VALUES ("ATTGGTTA", 2);
55 INSERT INTO BINDING (b_sequence, g_id) VALUES ("DDTTAAG", 3);
```