

Solutions exercises

SLIDE 3

- Find the gene named *MALAT1* in the gene table (bioinf_testdb)
 - In addition to the chromosomal position, return the size of the gene

```
SELECT
    chromosome,
    seq_region_start,
    seq_region_end,
    (seq_region_end - seq_region_start + 1)
FROM
    bioinf_testdb.gene
WHERE
    gene_name = 'MALAT1';
```

- Find the known miRNA that is located the most distal on the p-arm of chromosome 2

```
SELECT
    *
FROM
    bioinf_testdb.gene
WHERE
    biotype = 'miRNA' and chromosome = 2 and status = 'KNOWN'
ORDER BY seq_region_start ASC
LIMIT 1;
```

- Find genes related to breast cancer (use description field)

```
SELECT
    *
FROM
    bioinf_testdb.gene
WHERE
    description LIKE "%breast cancer%";
```

- Return a list of genes located on chromosome Y in alphabetical order

```
SELECT
    *
FROM
    bioinf_testdb.gene
WHERE
    chromosome = 'Y'
ORDER BY gene_name ASC;
```

SLIDE 4

- Return a list of the number of genes per biotype, most abundant biotypes first

```
SELECT
    COUNT(*), biotype
FROM
    bioinf_testdb.gene
GROUP BY biotype
ORDER BY COUNT(*) DESC;
```

- **Return a list of the number of genes per status**

```
SELECT
  COUNT(*), status
FROM
  bioinf_testdb.gene
GROUP BY status
ORDER BY COUNT(*) DESC;
```

```
SELECT
  COUNT(*), biotype, status
FROM
  bioinf_testdb.gene
GROUP BY biotype, status
ORDER BY COUNT(*) DESC;
```

- **Select only those biotypes that cover at least 1% of the human genome (hint: 'size' in previous exercise) and return this percentage.**

```
SELECT
  SUM(seq_region_end - seq_region_start + 1) / 3000000000 as pc, biotype
FROM
  bioinf_testdb.gene
GROUP BY biotype
HAVING pc >= 0.01
ORDER BY pc DESC;
```

SLIDE 5

- **How many transcript does the PTEN gene have?**

```
SELECT
  COUNT(*)
FROM
  bioinf_testdb.gene
  JOIN
  transcript ON transcript.gene_id = gene.gene_id
WHERE
  gene_name = 'PTEN';
```

- **Return the position of the exons of transcript 221260**

```
SELECT
  exon.seq_region_start,
  exon.seq_region_end
FROM
  exon_transcript
  JOIN
  exon ON exon_transcript.exon_id = exon.exon_id
WHERE
  exon_transcript.transcript_id = 221260;
```

- **Return the transcripts of the *TP53* gene**
 - **Return their exons as well**
 - **Find the longest spliced transcript of *TP53* (taking into account the intron-exon structure)**
 - **How many exons does each transcript have?**

```
SELECT
  transcript.transcript_id
```

```

FROM
    bioinf_testdb.gene
    JOIN
        transcript ON transcript.gene_id = gene.gene_id
WHERE
    gene.gene_name = 'TP53';

```

```

SELECT
    transcript.transcript_id, exon.exon_id
FROM
    bioinf_testdb.gene
    JOIN
        transcript ON transcript.gene_id = gene.gene_id
    JOIN
        exon_transcript ON exon_transcript.transcript_id = transcript.transcript_id
    JOIN
        exon ON exon.exon_id = exon_transcript.exon_id
WHERE
    gene.gene_name = 'TP53';

```

```

SELECT
    transcript.transcript_id,
    SUM(exon.seq_region_end - exon.seq_region_start + 1) AS size
FROM
    bioinf_testdb.gene
    JOIN
        transcript ON transcript.gene_id = gene.gene_id
    JOIN
        exon_transcript ON exon_transcript.transcript_id = transcript.transcript_id
    JOIN
        exon ON exon.exon_id = exon_transcript.exon_id
WHERE
    gene.gene_name = 'TP53'
GROUP BY transcript.transcript_id
ORDER BY size DESC
LIMIT 1;

```

```

SELECT
    transcript.transcript_id,
    COUNT(*) AS nr_exons,
    SUM(exon.seq_region_end - exon.seq_region_start + 1) AS size
FROM
    bioinf_testdb.gene
    JOIN
        transcript ON transcript.gene_id = gene.gene_id
    JOIN
        exon_transcript ON exon_transcript.transcript_id = transcript.transcript_id
    JOIN
        exon ON exon.exon_id = exon_transcript.exon_id
WHERE
    gene.gene_name = 'TP53'
GROUP BY transcript.transcript_id
ORDER BY size DESC;

```

→ Combined query

```

SELECT
    gene.gene_id,
    transcript.transcript_id,
    COUNT(*) as nr_exons,

```

```

SUM(exon.seq_region_end-exon.seq_region_start+1) as size
FROM
  bioinf_testdb.gene
  JOIN
  transcript ON transcript.gene_id = gene.gene_id
  JOIN
  exon_transcript ON exon_transcript.transcript_id = transcript.transcript_id
  JOIN
  exon ON exon.exon_id = exon_transcript.exon_id
WHERE
  gene.gene_name = 'TP53'
GROUP BY
  transcript.transcript_id
ORDER BY
  size DESC;

```

SLIDE 6

- **What is the name of the gene associated with transcript 260392?**

```

SELECT
  gene.gene_name
FROM
  transcript
  JOIN
  gene ON gene.gene_id = transcript.gene_id
WHERE
  transcript_id = 260392;

```

- **A mutation was found on chromosome 20, position 44002590. Which gene(s) overlap(s) with this position?**

```

SELECT
  gene_name
FROM
  gene
WHERE
  chromosome = '20'
  AND seq_region_start < 44002590
  AND seq_region_end > 44002590;

```

- **Select only those genes that have exons that overlap with this mutation. Which genes are they?**

```

SELECT
  gene.gene_name, COUNT(*) AS total_exons
FROM
  gene
  JOIN
  transcript ON gene.gene_id = transcript.gene_id
  JOIN
  exon_transcript ON exon_transcript.transcript_id = transcript.transcript_id
  JOIN
  exon ON exon.exon_id = exon_transcript.exon_id
WHERE
  gene.chromosome = '20'
  AND exon.seq_region_start < 44002590
  AND exon.seq_region_end > 44002590
GROUP BY gene.gene_name;

```

- **A biotype column can be found in both the gene and transcript table. Are there transcripts that have a different biotype from the gene they're part of? What are their names?**

```

SELECT

```

```

COUNT(*)
FROM
  gene
  JOIN
  transcript ON gene.gene_id = transcript.gene_id
WHERE
  gene.biotype != transcript.biotype;

```

- **Does the same go for status?**

```

SELECT
  COUNT(*)
FROM
  gene
  JOIN
  transcript ON gene.gene_id = transcript.gene_id
WHERE
  gene.status != transcript.status;

```

- **Which chromosome has the most genes and how many are there?**

```

SELECT
  chromosome, COUNT(*) AS total_genes
FROM
  gene
GROUP BY chromosome
ORDER BY total_genes DESC
LIMIT 1;

```

- **Which exon is the largest in the genome and how many base pairs are there?**

```

SELECT
  exon_id, (seq_region_end - seq_region_start + 1) AS length
FROM
  exon
ORDER BY length DESC
LIMIT 1;

```

- **Which transcript has the most exons and how many are there?**

```

SELECT
  transcript.transcript_id, COUNT(*) AS total_exons
FROM
  transcript
  JOIN
  exon_transcript ON exon_transcript.transcript_id = transcript.transcript_id
  JOIN
  exon ON exon_transcript.exon_id = exon.exon_id
GROUP BY transcript.transcript_id
ORDER BY total_exons DESC
LIMIT 1;

```

Rehearsal exercise

```

CREATE TABLE `trainings` (
  `training_id` INT NOT NULL,
  `subject` VARCHAR(50) DEFAULT NULL,
  `duration` INT(5) NOT NULL,
  PRIMARY KEY (`training_id`)
);

```

```

CREATE TABLE `lab_members` (
  `members_id` INT NOT NULL,
  `firstname` VARCHAR(50) DEFAULT NULL,

```

```

        `lastname` VARCHAR(50) DEFAULT NULL,
        `birth_date` DATE DEFAULT NULL,
        PRIMARY KEY (`members_id`)
    );

CREATE TABLE `members_in_training` (
    `mit_id` INT NOT NULL,
    `training_id` INT NOT NULL,
    `members_id` INT NOT NULL,
    FOREIGN KEY (`training_id`)
        REFERENCES trainings (`training_id`),
    FOREIGN KEY (`members_id`)
        REFERENCES lab_members (`members_id`)
    );

```

```

CREATE TABLE `equipment` (
    `equipment_id` INT NOT NULL,
    `eq_name` VARCHAR(50) DEFAULT NULL,
    `manufacturer` VARCHAR(50) DEFAULT NULL,
    `purchase_date` DATE DEFAULT NULL,
    PRIMARY KEY (`equipment_id`)
    );

```

```

CREATE TABLE `experiment` (
    `experiment_id` INT NOT NULL,
    `ex_name` VARCHAR(50) DEFAULT NULL,
    `performed_by` INT NOT NULL,
    `equipment_id` INT NOT NULL,
    `date` DATE DEFAULT NULL,
    PRIMARY KEY (`experiment_id`),
    FOREIGN KEY (`equipment_id`)
        REFERENCES equipment (`equipment_id`),
    FOREIGN KEY (`performed_by`)
        REFERENCES lab_members (`members_id`)
    );

```

```

CREATE TABLE `results` (
    `results_id` INT NOT NULL,
    `directory` VARCHAR(100) DEFAULT NULL,
    `experiment` INT NOT NULL,
    `status` ENUM('FAILED','PROGRESS','COMPLETED') DEFAULT NULL,
    PRIMARY KEY (`results_id`),
    FOREIGN KEY (`experiment`)
        REFERENCES experiment (`experiment_id`)
    );

```

-- INSERT SOME DATA

```

INSERT INTO trainings VALUES (1,'titreren', '2'),(2, 'afwegen', 5);
INSERT INTO trainings VALUES (3,'microscopy',7), (4,'cleaning',6), (5,'waste',5);
INSERT INTO trainings VALUES
(6,'pipetteren',15),(7,'microscopie',14),(8,'bloedprikken',21),(9,'enten',5),(10,'WASP-
toestel',28);

```

```

INSERT INTO lab_members VALUES ('1','Niels','Vanneste','1992-11-
26'),('2','Piet','Verkest','1989-04-23'),(3, 'Jos', 'Vermeulen', '1970-04-12');
INSERT INTO lab_members VALUES (4,'Pieterjan','De Coninck','1995-04-12'),
(5,'John','Doe','1991-10-02'), (6,'Mel','Trotter','1991-04-11'), (7,'Bill','Schuette','1970-12-01');

```

```
INSERT INTO lab_members VALUES (8,'Dries','Demeester','1993-02-27'),(9,'Bert','Demeester','1995-03-02'),(10,'Thijs','Demeester','1998-10-27'),(11,'Johan','Demeester','1960-04-08'),(12,'Myriam','Degezelle','1956-11-10');
```

```
INSERT INTO members_in_training VALUES (1, 1, 1),(2,2,2),(3,1,3);
INSERT INTO members_in_training VALUES (3,3,4), (4,4,7);
INSERT INTO members_in_training VALUES
(5,1,1),(6,1,3),(7,1,5),(8,2,4),(9,2,1),(10,3,4),(11,3,3),(12,3,2),(13,3,1),(14,4,5),(15,5,3),(16,5,2);
```

```
INSERT INTO equipment VALUES ('1', 'buret','duran', '2016-11-20'),(2,'erlenmeyer','pyrex','2016-10-23');
INSERT INTO equipment VALUES (3,'microscope','Bio-Rad','2015-03-02'),
(4,'pipet','Roche','2016-08-24');
INSERT INTO equipment VALUES (5,'WASP','Bio-Rad','2016-05-08'),(6,'entnaalden','Roche','2016-11-16');
```

```
INSERT INTO experiment VALUES (1, 'titratie', 1, 1, '2016-12-14'),(2,'wegen', 2, 2,'2016-11-03'),(3, 'titratie', 3, 1, '2010-05-12');
INSERT INTO experiment VALUES (4,'Salmonella',4,3,'2016-09-15'), (5,'Quality',7,4,'2013-12-03');
INSERT INTO experiment VALUES (6,'kwaliteitscontrole',4,6,'2016-11-20'),(7,'WASP_validatie',1,5,'2016-05-09'),(8,'microscoop_onderhoud',2,3,'2016-12-14'),(9,'pipetcontrole',5,4,'2015-12-28'),(10,'pipetteervaardigheid',3,4,'2014-02-15');
```

```
INSERT INTO results VALUES (1,
'home/results/titratie',1,'COMPLETED'),(2,'home/resulsts/balans', 2, 'PROGRESS'), (3,
'home/results/titratie', 1, 'FAILED');
INSERT INTO results VALUES (4,'Bacteria',4,'COMPLETED'), (5,'QC',5,'FAILED');
INSERT INTO results VALUES
(6,'bacteriologie',1,'COMPLETED'),(7,'onderhoud',3,'PROGRESS'),(8,'controles',4,'COMPLET
ED'),(9,'testen',5,'COMPLETED'),(10,'validaties',2,'COMPLETED');
```

- **All experiment equipment purchased after 1st of January 1985**

```
SELECT
    eq_name, purchase_date, ex_name
FROM
    equipment eq
    JOIN
    experiment ex ON eq.equipment_id = ex.equipment_id
WHERE
    purchase_date > '1985-01-01';
```

- **The number of experiments each lab member conducted**

```
SELECT
    CONCAT(firstname, ' ', lastname) AS name,
    COUNT(*) AS number_of_experiments
FROM
    experiment ex
    JOIN
    lab_members lm ON ex.performed_by = lm.members_id
GROUP BY name;
```

- **A list with all equipment used in a successful experiment**

```
SELECT DISTINCT
    (eq_name), status, ex_name
```

```

FROM
  results r
  JOIN
  experiment ex ON r.experiment = ex.experiment_id
  JOIN
  equipment eq ON ex.equipment_id = eq.equipment_id
WHERE
  status = 'COMPLETED';

```

- **A list with all lab members that failed an experiment**

```

SELECT
  CONCAT(firstname, ' ', lastname) AS name, ex_name, status
FROM
  results r
  JOIN
  experiment ex ON r.experiment = ex.experiment_id
  JOIN
  lab_members lm ON ex.performed_by = lm.members_id
WHERE
  status = 'FAILED';

```

- **Who followed wich trainings?**

```

SELECT
  CONCAT(firstname, ' ', lastname) AS name, subject
FROM
  lab_members lm
  JOIN
  members_in_training mit ON lm.members_id = mit.members_id
  JOIN
  trainings t ON mit.training_id = t.training_id;

```

- **Number of participants per training**

```

SELECT
  subject, COUNT(*) AS number_of_participants
FROM
  trainings t
  JOIN
  members_in_training mit ON t.training_id = mit.training_id
  JOIN
  lab_members lm ON mit.members_id = lm.members_id
GROUP BY subject;

```