

BIT05 – Database technologies

Jasper Anckaert

Lecture 3 – Database schema

Previously

- JOIN
`SELECT * FROM tbl1 JOIN tbl2 ON tbl1.col1 = tbl2.col2;`
- INNER, LEFT, RIGHT, OUTER
- Foreign key
 - Primary key of other table
 - Index
- Relations between tables
 - 1:n one-to-many relationship
 - n:m many-to-many relationship (xref-table)
- Views
`CREATE VIEW viewname as SELECT ...`
- Index
 - 1 per query

Previously

- Allow redundancy

SNOWFLAKE

- No redundancy
- Easy to maintain and change
- Complex queries
- Slower (more JOINs)
- Uses less space
- Bottom up

STAR

- Redundant data
- Less easy to maintain/change
- Lower query complexity
- Faster
- Uses more space (data is stored twice or more)
- Top down

- DUMP

- Create database backup

Relational databases with MySQL

Rehearsal exercises

- Create a new database for your lab and include following data
 - All trainings
 - Subject, duration
 - All lab members
 - Name, lastname, birth_date, training
 - All equipment
 - Name, manufacturer, purchase_date
 - All experiments
 - Name, performed_by, equipment_used, date
 - All results
 - Directory, experiment, status
- Fill with some data

```
CREATE TABLE training
  `id_training` int(11) NOT NULL,
  `subject` varchar(45) NOT NULL,
  `duration` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`id_member`)
  FOREIGN KEY ('')
```

```
CREATE TABLE lab_members
  `id_member` int(11) NOT NULL,
  `name` varchar(45) NOT NULL,
  `lastname` varchar(45) NOT NULL,
  `birth_date` date() NOT NULL,
  `training`
  PRIMARY KEY (`id_member`)
  FOREIGN KEY ('')
```

```
CREATE TABLE equipment
  `id_equipment` int(11) NOT NULL,
  `name` varchar(45) NOT NULL,
  `manufacturer` varchar(45) NOT NULL,
  `purchase_date` date() NOT NULL,
  PRIMARY KEY (`id_member`)
  FOREIGN KEY ('')
```

```
CREATE TABLE experiments
  `id_experiments` int(11) NOT NULL,
  `name` varchar(45) NOT NULL,
  `performed_by` varchar(45) NOT NULL,
  `equipment_used` varchar(45) NOT NULL,
  `date` date() NOT NULL,
  PRIMARY KEY (`id_member`)
  FOREIGN KEY ('')
```

```
CREATE TABLE results
  `id_results` int(11) NOT NULL,
  `experiment` varchar(45) NOT NULL,
  `status` varchar(45) NOT NULL,
  PRIMARY KEY (`id_member`)
  FOREIGN KEY ('')
```

```
CREATE TABLE members_equipment_experiments (
  id INT(9) AUTO_INCREMENT NOT NULL,
  lab_member INT (9),
  equipment INT(9),
  experiment INT(9),
  PRIMARY KEY (id),
  FOREIGN KEY (equipment)
  REFERENCES equipment(equipment_id),
  FOREIGN KEY (experiment)
  REFERENCES experiments(experiment_id),
  FOREIGN KEY (lab_member)
  REFERENCES lab_members(member_id)
);
INSERT INTO members_equipment_experiments (lab_member, equipment, experiment) VALUES
  (1,1,1),(1,1,2),(1,2,2),
  (2,2,2),(2,2,3),(2,3,3),
  (3,3,3),(3,3,4),(3,4,4),
  (4,4,4),(4,4,2),(4,5,2),
  (5,5,2),(5,2,2),(5,2,3),
  (6,6,2),(6,6,1),(6,6,2),
  (7,1,1),(7,5,4),(7,4,3),
  (8,2,3),(8,1,4),(8,3,3),
  (9,2,1),(9,3,3),(9,5,4)
;
```

```
CREATE TABLE All_trainings (
    `Subject` VARCHAR(255),
    `Duration` INT);

CREATE TABLE ALL_lab_members (
    `Name` VARCHAR(255),
    `Lastname` VARCHAR(255),
    `Birth_date` date,
    `Training` VARCHAR(255),
    PRIMARY KEY (Name));

CREATE TABLE All_equipment (
    `Name` VARCHAR(255),
    `Manufacturer` VARCHAR(255),
    `Purchase_date` date,
    PRIMARY KEY (Name));

CREATE TABLE All_experiments (
    `Name` VARCHAR(255),
    `Performed_by` VARCHAR(255),
    `Equipment_used` VARCHAR(255),
    FOREIGN KEY (Name)
    REFERENCES All_equipment (Name),
    FOREIGN KEY (Name)
    REFERENCES ALL_equipment (Name));

CREATE TABLE All_results (
    `Directory` VARCHAR(255),
    `Experiment` VARCHAR(255),
    `Status` VARCHAR(255),
    FOREIGN KEY (Name)
    REFERENCES All_experiments (Name));
```

```
CREATE TABLE `training` (
    train_id int(11) NOT NULL,
    subject varchar(45),
    duration varchar(45) DEFAULT NULL,
    PRIMARY KEY(train_id),
);

CREATE TABLE `lab_members` (
    mem_id int(11) NOT NULL,
    name varchar(45),
    lastname varchar(45),
    birth_date date(YYYY-MM-DD),
    training varchar(45),
    PRIMARY KEY(mem_id),
);

CREATE TABLE `equipments` (
    eq_id int(11) NOT NULL,
    name varchar(45),
    manufacturer varchar(45),
    purchase_date date(YYYY-MM-DD),
    PRIMARY KEY(eq_id),
);

CREATE TABLE `experiments` (
    exp_id int(11) NOT NULL,
    name varchar(45),
    performed_by varchar(45),
    equipment_used varchar(45),
    date_performed date(YYYY-MM-DD),
    PRIMARY KEY(exp_id),
    FOREIGN KEY (perfomed_by),
    references 'lab_members',
);

CREATE TABLE `results` (
    res_id int(11),
    directory varchar(45),
    experiment varchar(45),
    status varchar(45),
    PRIMARY KEY(res_id),
    FOREIGN KEY(experiment),
    references 'experiment',
);
```

```
create table trainings (
    training_id int(11) UNIQUE NOT NULL,
    subject enum('PCR', 'qPCR', 'NGS', 'FRAGMENTANALYSIS'),
    duration_days int(5),
    PRIMARY KEY (training_id),
);
```

```
CREATE TABLE `trainings` (
    `id` int(11) NOT NULL,
    `subject` varchar(45) DEFAULT NULL,
    `duration` int(10) DEFAULT NULL,
    PRIMARY KEY (`id`),
    FOREIGN KEY (`subject`) REFERENCES lab_members(`subject`)
);

INSERT INTO `trainings` VALUES (1,'pipeteren',60),(2,'Fungiexpressicobeheer',240);
```

```
CREATE TABLE `lab_members` (
    `member_id` int(11) NOT NULL,
    `name` varchar(45) DEFAULT NULL,
    `surname` varchar(45) DEFAULT NULL,
    `birthdate` datetime DEFAULT NULL,
    `subject` varchar(45) DEFAULT NULL,
    `lab` int(10) DEFAULT NULL,
    PRIMARY KEY (`member_id`),
    FOREIGN KEY (`subject`) REFERENCES trainings(`subject`)
);
```

```
create table trainings(
tr_id char(10) auto_increment not null,
subject varchar(255),
duration_weeks int(5),
PRIMARY KEY(tr_id)
);
INSERT INTO 'trainings' VALUES(1, 'PCR', 1),(2, 'PCR', 2),(3, 'qPCR', 1),(4, 'RNA extraction', 2),
(5,'ION torrent',1),(6,'gel filtration',2),(7, 'purification', 1),(8, 'Sequencing', 2);
```

```
CREATE TABLE `training` (training_id int not null,
subject varchar(255) ,
duration varchar(255),
primary key (training_id)
);
CREATE TABLE `lab_members` (lab_member_id int not null,
name varchar(255),
lastname varchar(255) ,
birth_date date default null ,
training varchar(255),
primary key (lab_member_id),
);
insert into `training` (training_id, subject, duration) values( '','lignin quantification and cell wall analysis','4 months'),( '','quality control of sugar and starches ','2 weeks');
```

```
CREATE TABLE `Lab_members` (
`id_member` int(9)auto_increment,
`name` varchar(20) DEFAULT NULL,
`lastname` varchar(20) DEFAULT NULL,
`birth_date` date DEFAULT NULL,
PRIMARY KEY (`id_member`)
);
```

```
Create table trainings (
    id int(11) UNIQUE NOT NULL,
    subject enum('HPLC','GC','MS'),
    duration FLOAT,
    PRIMARY KEY (id)
);
```

```
create table lab_members(
    id int(11) UNIQUE NOT NULL,
    name varchar(255) NOT NULL,
    lastname varchar(255),
    birth_date datetime,
    PRIMARY KEY (id)
);
```

```
create table trainings_lab_members (
    id int(11) UNIQUE NOT NULL,
    lab_member int(11),
    training int(11),
    PRIMARY KEY (id),
    FOREIGN KEY (training)
    REFERENCES trainings(id),
    FOREIGN KEY (lab_member)
    REFERENCES lab_members(id)
);
```

```
CREATE TABLE `Training` (
    `Subject` varchar(255) not null,
    `Duration_in_hour` int(5) default null,
    primary key (`Subject`),
    key (`Subject`)
);
```

```
CREATE TABLE experiments (
    experiment_id int(5) NOT NULL,
    name varchar (20) DEFAULT NULL,
    labmember_id int(5) DEFAULT NULL,
    equipment_id int(5) DEFAULT NULL,
    date date DEFAULT NULL,
    PRIMARY KEY(experiment_id)
);
```

```
CREATE TABLE 'All_trainings' (
    subject int(7) NOT NULL,
    duration int(7) NOT NULL,
    PRIMARY KEY (subject),
    );
```

```
CREATE TABLE 'All_lab_members' (
    name VARCHAR(45) DEFAULT NULL,
    lastname VARCHAR(45) DEFAULT NULL,
    birth_date date DEFAULT NULL,
    training int(7) DEFAULT NULL,
    PRIMARY KEY (name),
    FOREIGN KEY (training)
    REFERENCES All_trainings(subject),
    );
```

```
CREATE TABLE trainings (
`training_id` int(7) NOT NULL,
`subject` varchar(45) DEFAULT NULL,
`duration` varchar(45) DEFAULT NULL,
PRIMARY KEY (`training_id`));
```

```
CREATE TABLE lab_members (
`member_id` int(7) NOT NULL,
`name` varchar(45) DEFAULT NULL,
`lastname` varchar(45) DEFAULT NULL,
`birth_date` date DEFAULT NULL,
`training` int(7) DEFAULT NULL,
PRIMARY KEY (`member_id`),
FOREIGN KEY (training)
REFERENCES trainings(training_id));
```

```
CREATE TABLE equipment(
`equipment_id` int(7) NOT NULL,
`name` varchar(45) DEFAULT NULL,
`manufacturer` varchar(45) DEFAULT NULL,
`purchase_date` date DEFAULT NULL,
PRIMARY KEY (`equipment_id`));
```

```
CREATE TABLE experiments (
`experiment_id` int(7) NOT NULL,
`name` varchar(45) DEFAULT NULL,
`performed_by` int(7) DEFAULT NULL,
`equipment_used` int(7) DEFAULT NULL,
`date` date DEFAULT NULL,
PRIMARY KEY (`experiment_id`),
FOREIGN KEY (equipment_used)
REFERENCES equipment(equipment_id),
FOREIGN KEY (performed_by)
REFERENCES lab_members(member_id));
```

```
CREATE TABLE results (
`result_id` int(7) NOT NULL,
`directory` varchar(45) DEFAULT NULL,
`experiment` int(7) DEFAULT NULL,
`status` enum('SUCES!', 'FAIL...', 'In Progress') DEFAULT NULL,
PRIMARY KEY (`result_id`),
FOREIGN KEY (experiment)
REFERENCES experiments(experiment_id));
```

```
CREATE TABLE members (
id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
name VARCHAR(30),
lastname VARCHAR(45),
birth_date date
);
```

```
CREATE TABLE training (
training_by INT,
subject VARCHAR(30),
duration TIME,
FOREIGN KEY (training_by) REFERENCES members(id)
);
```

```
CREATE TABLE All_trainings(
Training_ID VARCHAR(6) NOT NULL,
Topic VARCHAR(255) DEFAULT NULL,
Duration TIME DEFAULT NULL,
PRIMARY KEY (Training_ID)
):
```

```

DROP TABLE IF EXISTS `Training`;
CREATE TABLE `Training` (
  `train_id` int(11) NOT NULL,
  `subject` varchar(100) DEFAULT NULL,
  `duration_days` int(4) DEFAULT NULL,
  PRIMARY KEY (`train_id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

DROP TABLE IF EXISTS `Lab_members`;
CREATE TABLE `Lab_members` (
  `member_id` int(11) NOT NULL,
  `name` varchar(45) DEFAULT NULL,
  `lastname` varchar(45) DEFAULT NULL,
  `birth_date` date DEFAULT NULL,
  `training` int(11) DEFAULT NULL,
  PRIMARY KEY (`member_id`),
  FOREIGN KEY (`training`)
    REFERENCES (`train_id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

DROP TABLE IF EXISTS `Equipment`;
CREATE TABLE `Equipment` (
  `equipm_id` int(11) NOT NULL,
  `equipment` varchar(100) DEFAULT NULL,
  `manufacturer` varchar(45) DEFAULT NULL,
  `purchase_date` date DEFAULT NULL,
  PRIMARY KEY (`equipm_id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

DROP TABLE IF EXISTS `Experiments`;
CREATE TABLE `Experiments` (
  `exp_id` int(11) NOT NULL,
  `title` varchar(250) DEFAULT NULL,
  `performed_by` int(11) DEFAULT NULL,
  `exp_date` date DEFAULT NULL,
  `equipment_used` int(11) DEFAULT NULL,
  PRIMARY KEY (`exp_id`),
  FOREIGN KEY (`performed_by`)
    REFERENCES (`member_id`),
  FOREIGN KEY (`equipment_used`)
    REFERENCES (`equipm_id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

DROP TABLE IF EXISTS `Results`;
CREATE TABLE `Results` (
  `result_id` int(11) NOT NULL,
  `directory` varchar(45) DEFAULT NULL,
  `experiment` int(11) DEFAULT NULL,
  `status` ENUM('FAILED', 'COMPLETED', 'PLANNED'),
  PRIMARY KEY (`result_id`),
  FOREIGN KEY (`experiment`)
    REFERENCES (`exp_id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

```

CREATE TABLE `Trainings` (
  `Training_ID` int(6) NOT NULL,
  `Subject` VARCHAR(255) DEFAULT NULL,
  `Duration_(min)` int(7) NOT NULL,
  `Members_in_training` VARCHAR(255) DEFAULT NULL,
  PRIMARY KEY (`Training_ID`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

DROP TABLE if exists `Lab_members`;
CREATE TABLE `Lab_members` (
  `Member_ID` int(6) NOT NULL,
  `Name` VARCHAR(255) Default null,
  `Lastname` VARCHAR(255) DEFAULT NULL,
  `Birth_date` date DEFAULT NULL,
  `Training` VARCHAR(255) DEFAULT NULL,
  PRIMARY KEY (`Member_ID`),
  FOREIGN KEY (`Training`, `Lastname`)
    REFERENCES `Trainings`(`Subject`, `Members_in_training`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

DROP TABLE if exists `Equipment`;
CREATE TABLE `Equipment` (
  `Eq_ID` int(6) NOT NULL,
  `Name` VARCHAR(255) DEFAULT NULL,
  `Manufacturer` VARCHAR(255) DEFAULT NULL,
  `Purchase_date` date DEFAULT NULL,
  PRIMARY KEY (`Eq_ID`),
  FOREIGN KEY (`Name`)
    REFERENCES `Experiment`(`Equipment_used`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

DROP TABLE if exists `Experiments`;
CREATE TABLE `Experiments` (
  `Exp_ID` int(6) NOT NULL,
  `Experiment` VARCHAR(255) DEFAULT NULL,
  `Performed_by` VARCHAR(255) DEFAULT NULL,
  `Equipment_used` VARCHAR(255) DEFAULT NULL,
  `Date` date DEFAULT NULL,
  PRIMARY KEY (`Exp_ID`),
  FOREIGN KEY (`Experiment`)
    REFERENCES `Results`(`Experiment_name`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

```
Create table All_trainings(
subjects VARCHAR(100),
duration int),
FOREIGN KEY (subjects),
REFERENCES All_lab_members(training);

INSERT INTO All_trainings
    (subjects, duration)
VALUES
    ("moleculaire biologie ",15),
    ("analytische chemie",20),
    ("immunology",25),
    ("comperatieve", 30);

create TABLE All_lab_members(
name VARCHAR(100),
lastname VARCHAR(100),
birth_date DATE,
training VARCHAR(100),
FOREIGN KEY (name),
REFERENCES All_experiments(performed_by);

INSERT INTO All_lab_members
    (name, lastname, birth_date,training)
VALUES
    ("thibault", "Courtois", 1/01/1999),
    ("thomas", "Vermaelen", 1/02/1999),
    ("Joeri", "Tielemans", 1/03/1999),
    ("Adnan", "Januzaj", 1/04/1999),
    ("Thorgan", "Hazard", 1/05/1999),
    ("Dries", "Mertens", 1/06/1999),
    ("Axel", "Witsel", 1/07/1999);
```

```
CREATE TABLE all_trainings (
    subject VARCHAR(255),
    duration TIME);
CREATE TABLE lab_members (
    member_id,
    name VARCHAR(255),
    lastname VARCHAR(255),
    birth_date DATE,
    training VARCHAR(255));
CREATE TABLE all_equipment (
    eq_id,
    name VARCHAR(255),
    manufacturer VARCHAR(255),
    purchase_date DATE);
CREATE Table all_experiments (
    experiment_id
    experiment_name VARCHAR(255),
    member_id
    equipment_used
```

Relational databases with MySQL

Rehearsal exercises

- Pay attention to
 - Semi-colon after each statement, parameters of create statement between brackets
 - Order of table creation
 - Cross-reference table if needed
 - Primary key in each table (Add column if needed)
 - Correct usage of constraints (foreign keys, ...) and column types

THINK BEFORE YOU CREATE

Relational databases with MySQL

Rehearsal exercises (part 2)

- In your newly created database, search for
 - All experiment equipment purchased after 1st of January 1985
 - All lab members born in April
 - The number of experiments each lab member conducted
 - A list with all equipment used in a successful experiment
 - A list with all lab members that failed an experiment
 - Who followed which trainings?
 - Number of participants per training

MySQL Workbench

Database schema

- MySQL monitor to execute DDL commands
 - Servers
 - Advanced users
- GUI
 - HeidiSQL
 - MySQL Workbench

MySQL Workbench

Installation

- Available as for download (several operationg systems)
 - <http://dev.mysql.com/downloads/workbench/>
- To install DEB package

```
# dpkg -i package.deb
```
- To install RPM package

```
# rpm -Uvh package.rpm
```
- To install on Windows/Mac
 - Double click package.msi or package.dmg
- Or install native package from repository

```
# yum install mysql-workbench-community
```



Welcome to MySQL Workbench

1. Add connection (connection is saved and can be reused)

.. It allows you to design,

create and browse your database schemas, work with database objects and insert data as well as design and run SQL queries to work with stored data. You can also migrate schemas and data from other database vendors to your MySQL database.

[Discuss on the Forums >](#)

MySQL Connections

Filter connections

2. Choose a name for the database connection



Me... MySQL Connections

Setup New Connection

Connection Name: Type a name for the connection

Connection Method: Standard (TCP/IP) Method to use to connect to the RDBMS

Parameters SSL Advanced

Hostname: Port: Name or IP address of the server host - and TCP/IP port.

Username: Name of the user to connect with.

Password: The user's password. Will be requested later if it's not set.

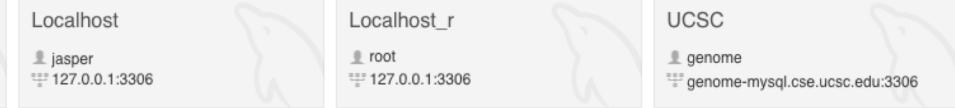
Default Schema: The schema to use as default schema. Leave blank to select it later.

Configure Server Management... Test Connection Cancel OK

5. Double click to open

3. Fill in connection/authentication parameters

4. Test and save the connection



This screenshot shows the MySQL Workbench interface with four main panes:

- Navigator pane:** Located on the left, it displays the database schema. It includes sections for MANAGEMENT, INSTANCE, PERFORMANCE, SCHEMAS, and a detailed view of the selected 'bioinf_testdb.gene' table.
- Query tabs:** Located at the top center, this pane is titled 'Query 1' and contains the SQL query: 'SELECT * FROM bioinf_testdb.gene;'. It also has a 'Limit to 1000 rows' option.
- Results pane:** Located in the center-right, it displays the results of the query in a grid format. The columns are: gene_id, biotype, chromosome, seq_region_start, seq_region_end, seq_region_strand, gene_name, source, status, and description. The results show 140 rows of data for lncRNA genes.
- Output pane:** Located at the bottom center, it provides an overview of recent queries. It lists the time, action (query), response (number of rows returned), and duration/fetch time for each query. The last three queries are shown, all returning 1000 rows with execution times around 0.0005 to 0.0006 seconds.

Annotations in teal text provide additional context:

- 'Query tabs: write SQL code here' points to the Query tabs pane.
- 'Results pane: the result of the query will be displayed here' points to the Results pane.
- 'Output pane: overview of the recent queries
If a query fails, the error message will be displayed here' points to the Output pane.
- 'Navigator pane: browse this database server' points to the Navigator pane.

Table Data (bioinf_testdb.gene):

gene_id	biotype	chromosome	seq_region_start	seq_region_end	seq_region_strand	gene_name	source	status	description
76121	lincRNA	15	32898619	32907361	-1	AC123768.6	ensembl	NOVEL	NULL
76122	lincRNA	15	35047285	35105124	1	AC087457.1	ensembl	KNOWN	NULL
76123	lincRNA	15	35838396	36151200	1	AC015994.1	ensembl	KNOWN	NULL
76124	lincRNA	15	37090798	37110707	-1	AC019016.1	ensembl	KNOWN	NULL
76125	lincRNA	15	38363831	38364884	-1	AC087473.1	ensembl	KNOWN	NULL
76126	lincRNA	15	40213243	40217099	1	AC012377.1	ensembl	NOVEL	NULL
76127	lincRNA	15	40331512	40359491	1	AC021755.4	ensembl	KNOWN	NULL
76128	lincRNA	15	41199009	41201535	1	AC025166.1	ensembl	NOVEL	NULL
76129	lincRNA	15	41576203	41601901	1	AC012652.1	ensembl	KNOWN	NULL
76130	lincRNA	15	43029401	43034389	1	AC090510.1	ensembl	KNOWN	NULL
76131	lincRNA	15	50647664	50650501	1	AC022087.1	ensembl	KNOWN	NULL
76132	lincRNA	15	51132073	51200796	-1	AC021752.1	ensembl	KNOWN	NULL
76133	lincRNA	15	52393790	52403225	1	AC023906.1	ensembl	KNOWN	NULL
76134	lincRNA	15	55609382	55611243	-1	AC018926.1	ensembl	KNOWN	NULL
76135	lincRNA	15	57592563	57599965	1	AC016525.1	ensembl	KNOWN	NULL
76136	lincRNA	15	59060273	59063173	-1	AC090515.1	ensembl	KNOWN	NULL
76137	lincRNA	15	60771380	60823166	1	AC087385.1	ensembl	KNOWN	NULL
76138	lincRNA	15	64663186	64667537	1	AC087632.1	ensembl	KNOWN	NULL
76139	lincRNA	15	66952560	66974114	1	AC013564.1	ensembl	KNOWN	NULL
76140	lincRNA	15	67332912	67340583	1	AC087482.1	ensembl	NOVEL	NULL

The screenshot shows the MySQL Workbench interface. On the left, the Schema browser displays a tree view of databases (biodb, bioinf, bioinf_examen, bioinf_testdb) and their tables (exon, exon_transcript, gene, transcript). The 'gene' table under 'bioinf_testdb' is selected and highlighted in grey. Below the schema browser, there are tabs for 'Object Info' and 'Session'. The 'Object Info' tab is active, showing detailed information about the 'gene' table. The table definition includes:

Column	Type	Properties
gene_id	int(10)	UN AI PK
biotype	varchar(40)	
chromo	varchar(40)	
some	varchar(40)	
seq_re	int(10)	UN
gion_st	int(10)	UN
art		
seq_reg	int(10)	UN
ion_end		
seq_reg	tinyint(2)	
ion_sra		
nd		
gene_n	varchar(128)	
ame		
source	varchar(20)	
status	enum('KNOWN','NOVEL','PUTATIVE','PREDICTED','UNKNOWN')	
descripti	text	
on		

Different databases on this server
BOLD = currently active database, all queries will be executed in this db, double click to change

Browse tables in the db here

More information on the currently selected item

MySQL Workbench

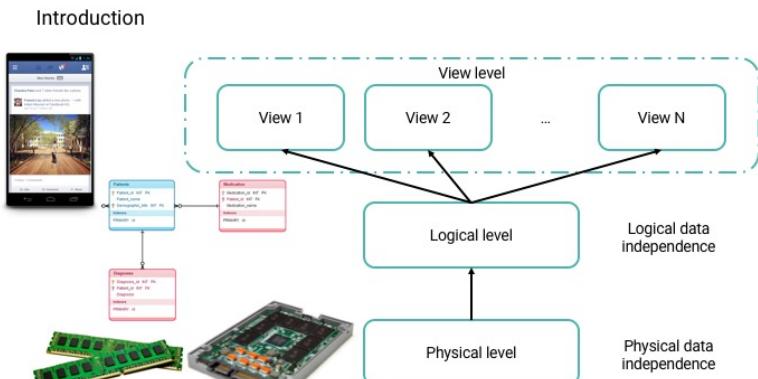
Exercices

- Connect to the MySQL database server
- Explore the server
 - How many databases are available to you?
 - How many tables does each database have?
 - What are the column types of the gene table (bioinf_testdb)?

MySQL Workbench

Data model

- Determines the structure of data
 - Conceptual data model
 - Structure of and relations between entities
 - No attributes or primary keys are specified
 - Logical data model
 - Attributes and primary key for each entity
 - Foreign keys are specified
 - Normalisation
 - Physical data model
 - Blueprint of the database
 - Physical means by which data are stored (partitions, CPUs, tablespaces, ...)
- Entity Relationship Diagram (ERD)



MySQL Workbench

Database models

- Flat model
 - Single two-dimensional array of data elements
 - E.g. spreadsheet
- Hierarchical model
 - Data is organized in a tree-like structure
 - Records are connected through links
- Network model
 - Each record can have multiple parents and child records

MySQL Workbench

Database models

- Relational model
 - Tables are relations
 - Links between tables are not explicitly defined → use keys
 - What we've been using so far but with deviations
- Object-relational model
 - Relational model with object-oriented features
 - PostgreSQL
- Object oriented model
 - Data is represented in the form of objects
 - Use same model of representation as in programming language

MySQL Workbench

Creating a database - Normalisation

- Organizing columns and tables
 - Reduce redundancy
 - Improve integrity
- Remember E. Codd?

MySQL Workbench

Normalisation

- UNF
 - Unnormalized form
 - Group all data in one entity
- 1NF
 - Eliminate repeating (and calculated) groups in individual tables
 - Create separate table for each set of related data
 - Identify each set of related data with a primary key
- 2NF
 - Every non-prime attribute of the table is dependent on the whole key of every candidate key
- 3NF
 - Every non-prime attribute is non-transitively dependent on every key

MySQL Workbench

Normalisation

- BCNF
 - Any attribute on which some other attribute is fully functionally dependent = determinant
 - Every determinant is a candidate key
- 4NF – 5NF – 6NF – DKNF

MySQL Workbench

Normalisation – example

Patient_no	Patient_name	Appointment_id	Time	Doctor
1	John	0	09:00	Zorro
2	Kerr	0	09:00	Killer
3	Adam	1	10:00	Zorro
4	Robert	0	13:00	Killer
5	Zane	1	14:00	Zorro

UNF DB(Patno,PatName,appNo,time,doctor)

1NF DB(Patno,PatName,appNo,time,doctor)

2NF DB(Patno,appNo,time,doctor)
R1(Patno,PatName)

3NF 2NF

BCNF DB(Patno,time,doctor)
R1(Patno,PatName)
R2(time,appNo)

MySQL Workbench

Normalisation – example

- Why is this table not in 1NF?
- Normalize up to 3NF
- Identify all keys in your 3NF relations

branchNo	branchAddress	telNos
B001	8 Jefferson Way, Portland, OR 97201	503-555-3618, 503-555-2727, 503-555-6534
B002	City Center Plaza, Seattle, WA 98122	206-555-6756, 206-555-8836
B003	14 – 8th Avenue, New York, NY 10012	212-371-3000
B004	16 – 14th Avenue, Seattle, WA 98128	206-555-3131, 206-555-4112

MySQL Workbench

Exercises

- Normalise up to 3NF

Pet_id	Pet_name	Pet_type	Pet_age	owner	Visit_date	procedure
246	Rover	dog	12	Sam Coock	2002-01-13	01 – Rabies vaccination
					2005-03-27	10 - Examination
					2003-04-02	05 – Heart worm test
296	Spot	dog	2	Terry Kim	2002-01-21	08 – Tetanus vaccination
					200-03-10	05 – Heart worm test
341	Morris	cat	4	Sam Coock	2001-01-23	01– Rabies vaccination
					2002-01-13	01 – Rabies vaccination
519	Tweedy	bird	2	Terry Kim	2002-04-30	20 – Check up
					2002-04-30	12 – Eye wash

MySQL Workbench

Exercises

- Normalise up to 3NF

INVOICE

HILLTOP ANIMAL HOSPITAL
INVOICE # 987

DATE: JAN 13/2002

MR. RICHARD COOK
123 THIS STREET
MY CITY, ONTARIO
Z5Z 6G6

PET	PROCEDURE	AMOUNT
ROVER	RABIES VACCINATION	30.00
MORRIS	RABIES VACCINATION	24.00
	TOTAL	54.00
	TAX (8%)	<u>4.32</u>
	AMOUNT OWING	<u>58.32</u>

MySQL Workbench

Exercises

- Normalise up to 3NF

Company	Symbol	Headquarters	Date	Close_price
Microsoft	MSFT	Redmond, WA	09/07/2013	23.96
Microsoft	MSFT	Redmond, WA	09/08/2013	23.93
Microsoft	MSFT	Redmond, WA	09/09/2013	24.01
Oracle	ORCL	Redwood Shores, CA	09/07/2013	24.27
Oracle	ORCL	Redwood Shores, CA	09/08/2013	24.14
Oracle	ORCL	Redwood Shores, CA	09/09/2013	24.33

MySQL Workbench

Exercises

- Normalise up to BCNF
 - Grade_report(StudNo,StudName,(Major,Adviser, (CourseNo,Ctitle,InstrucName,InstructLocn,Grade)))
 - Functional dependencies
 - StudNo \rightarrow StudName
 - CourseNo \rightarrow Ctitle,InstrucName
 - InstrucName \rightarrow InstrucLocn
 - StudNo,CourseNo,Major \rightarrow Grade
 - StudNo,Major \rightarrow Advisor
 - Advisor \rightarrow Major

MySQL Workbench

Exercises

video(title,director,serial)

customer(name,addr,memberno)

hire(memberno,serial,date)

title->director,serial

serial->title serial->director

name,addr -> memberno

memberno -> name,addr

serial,date -> memberno

- What normal form is this?
- Convert to BCNF

MySQL Workbench

Exercises

- Convert to BCNF

Drinkers(name, addr, beersLiked, manf, favBeer)

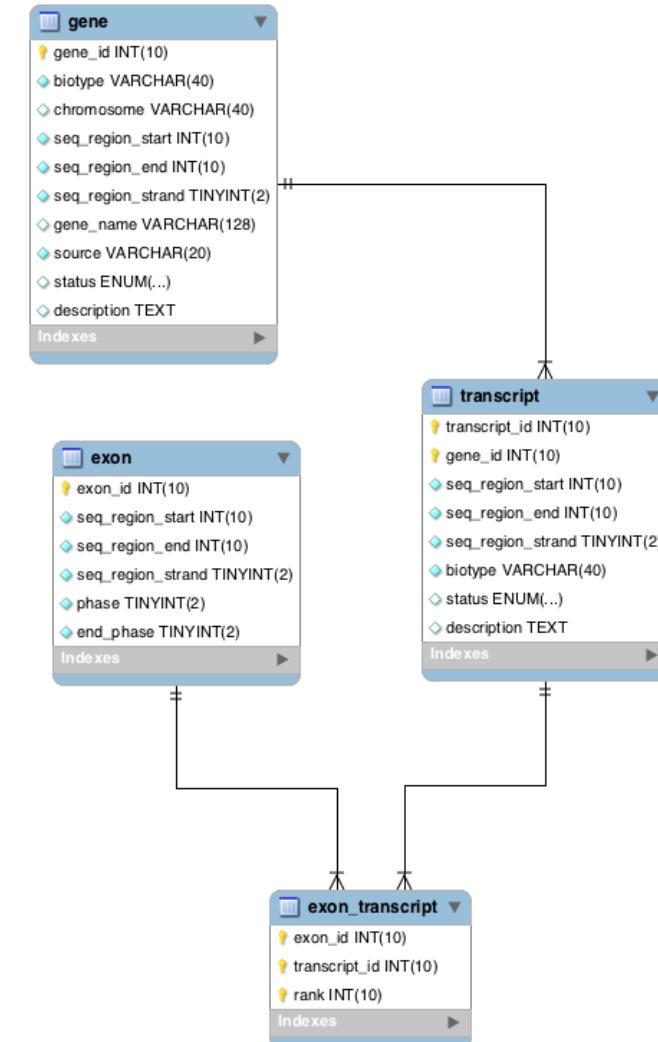
$\text{name} \rightarrow \text{addr, favBeer}$

$\text{beersLiked} \rightarrow \text{manf}$

MySQL Workbench

Creating tables

- Use the "model" interface in MySQL Workbench
 - DDL statements will be auto-generated
- Tables can be placed anywhere and dragged around
- Foreign keys will be displayed as lines and arrows





Models

Oplossing_model

 ...x (Personal)/Howest/examen
 mydb
 23 Jan 17, 15:07

Create a new database model

Untitled - MySQL Workbench

MySQL Model

Description

EER Diagrams

No Selection

Add Diagram

Physical Schemas

mydb MySQL Schema

Tables (0 items)

Add Table

Views (0 items)

Add View

Routines (0 items)

Add Routine

Routine Groups (0 items)

Add Group

User Types History

Type Definition Flags

Schema Privileges

SQL Scripts

Model Notes

Search

Templates

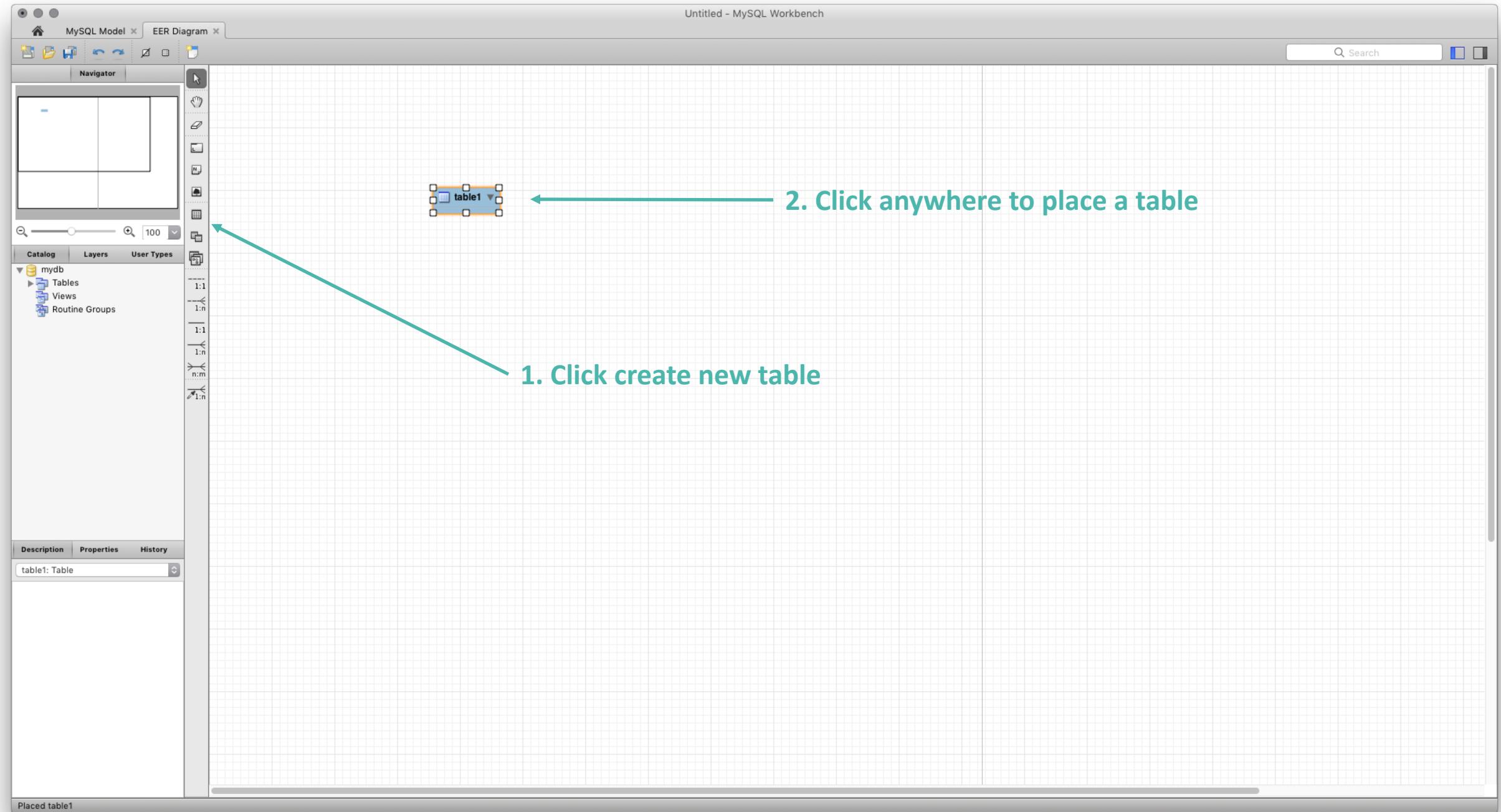
timestamps
create_time, update_time

user
username, email, password, crea...

category
category_id, name

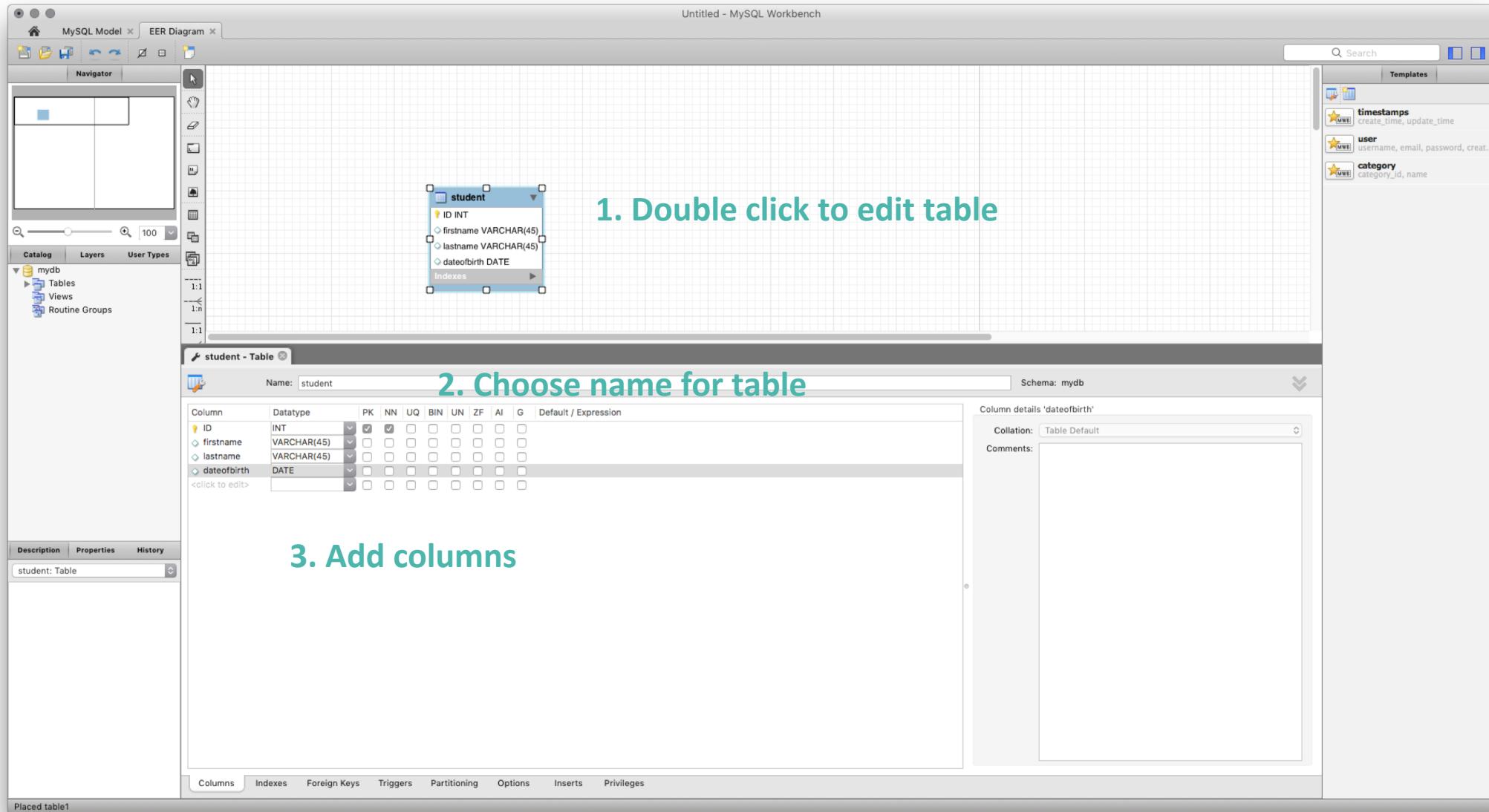
New document.

Add a new diagram



MySQL Workbench

Creating tables



MySQL Workbench

Add columns to table

UNSIGNED (for numeric types, eg. INT)

student - Table

Name: student

Column	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	G	Default / Expression
ID	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
firstname	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
lastname	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
dateofbirth	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<click to edit>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column name
Column type
Primary key
NOT NULL
UNIQUE
Auto increment

MySQL Workbench

Add foreign keys to table

The screenshot shows the MySQL Workbench interface with an EER Diagram and a Table Editor.

EER Diagram: Shows two tables: **student** and **study**. A dashed line connects them, indicating a relationship. The **student** table has columns: ID (PK), firstname, lastname, dateofbirth, and study_id. The **study** table has columns: ID (PK), title, kind, and indexes.

Table Editor (student - Table):

- Step 1:** The text "1. Make sure the table that holds the Foreign Key column is selected!" is overlaid on the diagram.
- Step 2:** The text "2. Switch to Foreign Keys tab" is overlaid on the Table Editor.
- Step 3:** The text "3. Add a new FK and select the referenced table" is overlaid on the Table Editor.

Foreign Key Details:

- Foreign Key:** student_study
- Referenced Table:** mydb.study
- Column:** study_id
- Referenced Column:** ID
- On Update:** NO ACTION
- On Delete:** NO ACTION

Annotations:

- A green arrow points from the text "Column on REFERENCED table (usually PK)" to the "ID" column in the "Referenced Column" list.
- A blue arrow points from the text "Column on THIS table" to the "study_id" column in the "Column" list.

MySQL Workbench

Exercises

- Create the following database schema

Students	
Student_number	
Name	
Last_name	
Birthdate	
Sex	
Trajectory_ID	



Course_of_student	
ID	
Student_ID	
Course_ID	



Courses	
ID	
Course	
Credits	

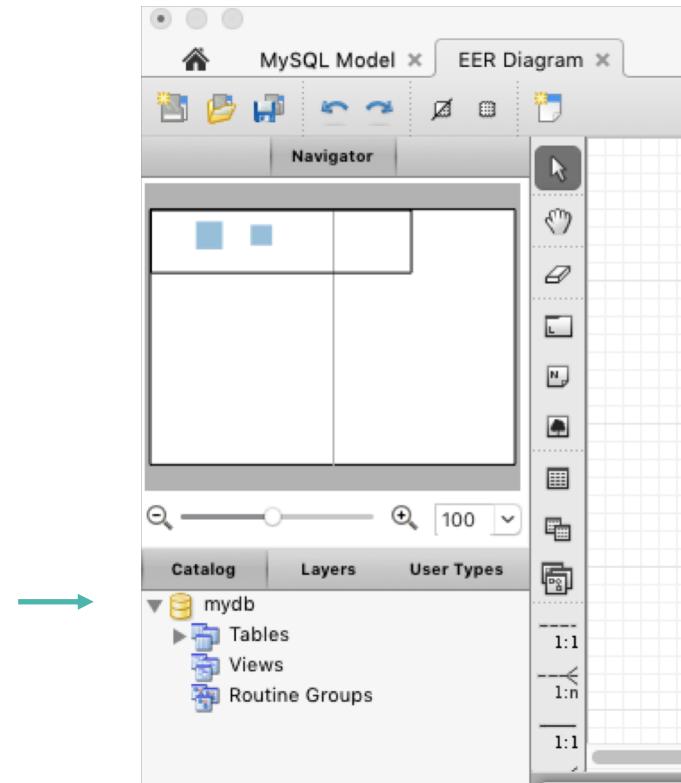
Trajectories	
ID	
Trajectory	

MySQL Workbench

Forward engineering

- Function in MySQL Workbench
 - Generates SQL code to create/modify a database based on your model
- Make sure the name of your database is correct!
 - Located under *Database > Forward engineer*
 - Check in the database browser (Refresh)

Double click
to change



MySQL Workbench

Exercises

- Create a MySQL table to track the movies you have watched:
 - Movie title
 - Genre: action, comedy, drama, horror, science fiction
 - Date you watched to movie
 - Score: 0-10
 - Comments
- Create a table to store your favourite directors and link it with the movie table
- Create a table to store your favourite actors and link it with the movie table
- Forward engineer your tables to your database
- Add some rows to the table you have created

MySQL Workbench

Creating a database

- Important questions
 - Which data?
 - Constraints?
 - Application?
 - Relations between data?



Entity relationship diagram

Relational databases with MySQL

Rehearsal exercises – a second try

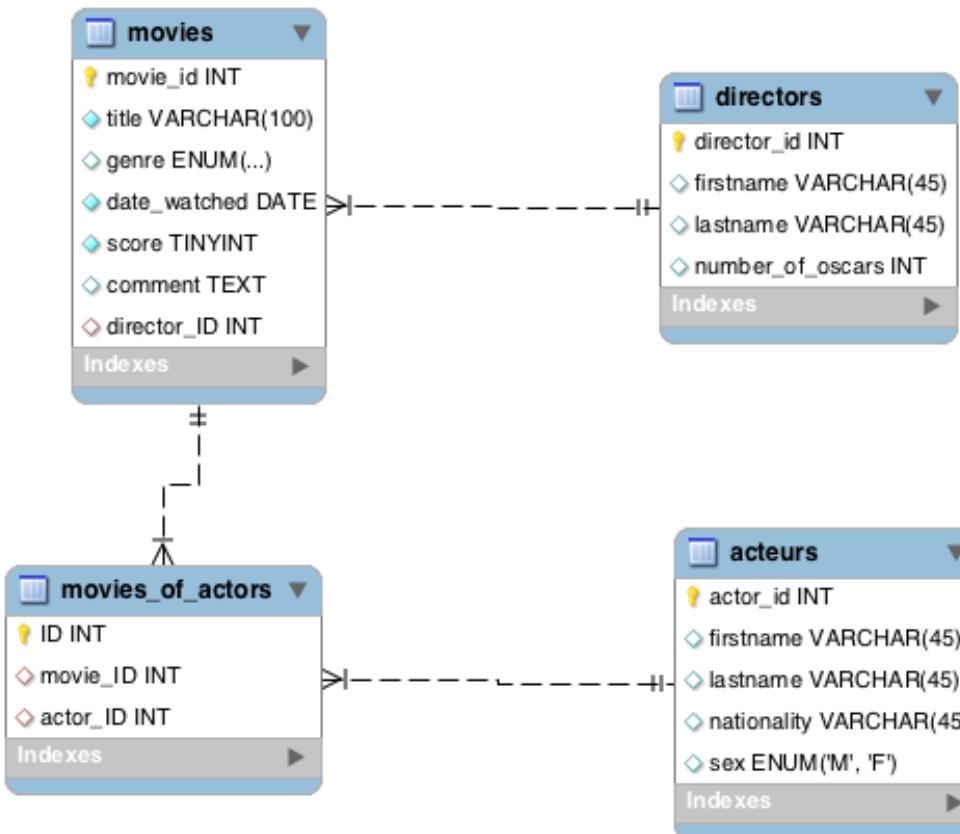
- Create a new database model for your lab and include following data
 - All trainings
 - Subject, duration
 - All lab members
 - Name, lastname, birth_date, training
 - All equipment
 - Name, manufacturer, purchase_date
 - All experiments
 - Name, performed_by, equipment_used, date
 - All results
 - Directory, experiment, status
- Forward engineer your model

*Be sure not to use the same
names as before if you want to
keep both databases*

MySQL Workbench

Reverse engineering

- Function in MySQL Workbench
 - Generates a (graphical) model of your database



MySQL Workbench

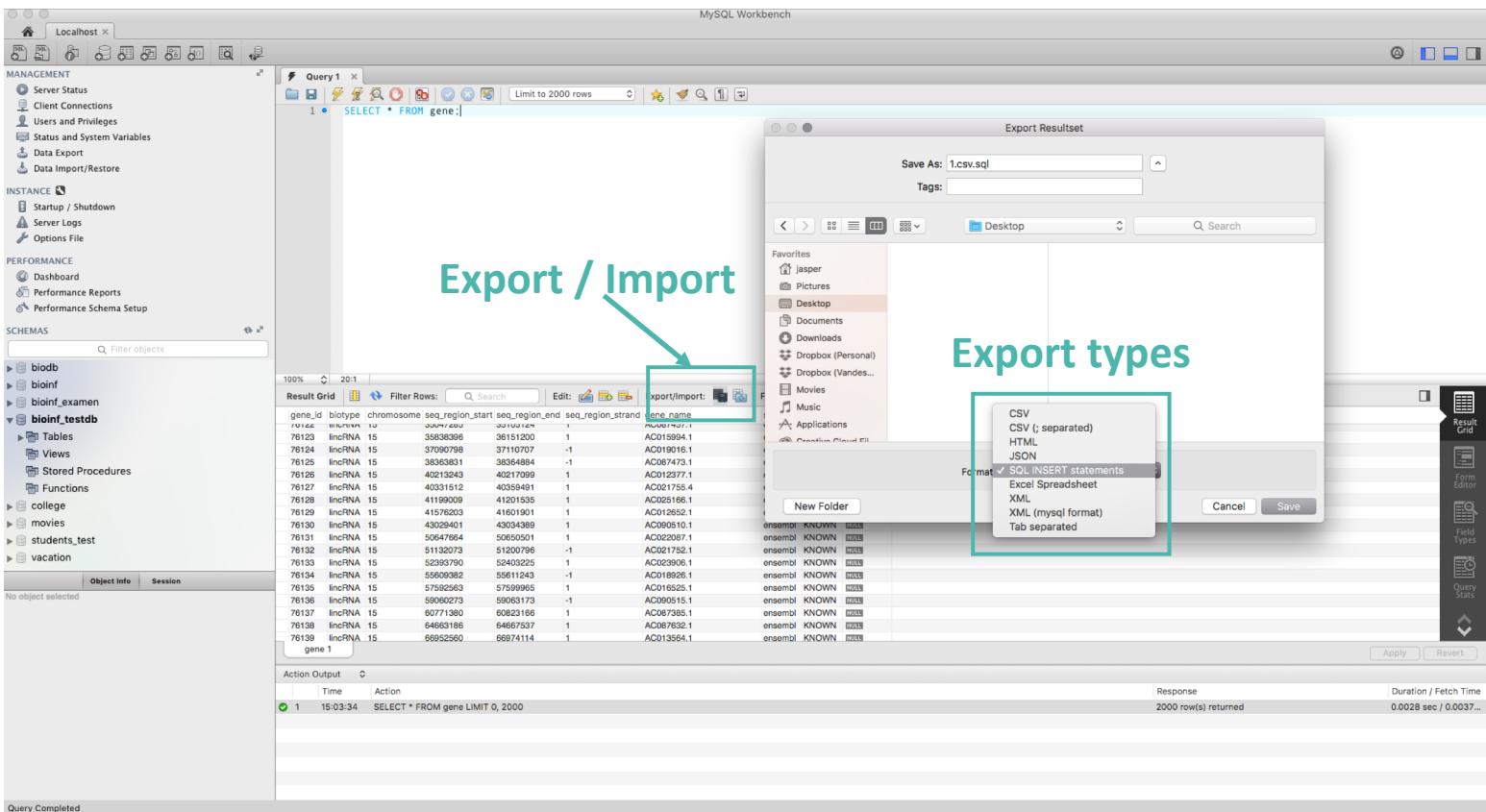
Exercises

- Reverse engineer the model of the bioinf_testdb
 - Check out the relationships between the different tables
 - Give the names of all the keys used

MySQL Workbench

Exporting data

- Remember
 \$ mysqldump [opt] db > db.sql
- Dump your database
 - Structure, data or both
 - Useful for backup
- Ability to export part of results (JSON, CSV, HTML, XML, ...)



MySQL Workbench

Import data

- Import entire dump file
- Import data from file (CSV, existing table, SQL, JSON)

The screenshot shows the MySQL Workbench interface with a result grid displaying two tables. The first table contains gene information with columns: gene_id, biotype, chromosome, seq_region_start, seq_region_end, seq_region_strand, and gene_name. The second table contains metadata with columns: source, status, and description. The 'Export/Import' icon in the toolbar is highlighted with a green box.

gene_id	biotype	chromosome	seq_region_start	seq_region_end	seq_region_strand	gene_name
70122	mRNA	19	30047200	30100124	1	AC0007407.1
76123	lincRNA	15	95838308	98151200	4	AC015004.1

source	status	description
ensembl	KNOWN	NULL
ensembl	KNOWN	NULL

MySQL Workbench

Excercises

- Export the data in the trajectories table
- Empty your table (TRUNCATE)
- Import data into the trajectories table using your export file

MySQL and other languages

PHP

- MySQLi extension
 - Works only with MySQL
 - Procedural vs. Object-Oriented
- PDO (PHP Data Objects)
 - Works with 12 different database systems

MySQL and other languages

PHP – MySQLi Procedural: Connect

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = mysqli_connect($servername, $username, $password);

// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
echo "Connected successfully";
?>
```

MySQL and other languages

PHP – MySQLi Procedural: Query

```
<?php  
  
// Create database  
$sql = "CREATE DATABASE myDB";  
if (mysqli_query($conn, $sql)) {  
    echo "Database created successfully";  
} else {  
    echo "Error creating database: " . mysqli_error($conn);  
}  
  
mysqli_close($conn);  
?>
```

- Other possibilities: INSERT, SELECT, DELETE, UPDATE

MySQL and other languages

PHP – MySQLi Procedural: INSERT

```
<?php

$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";

if (mysqli_query($conn, $sql)) {
    $last_id = mysqli_insert_id($conn);
    echo "New record created successfully. Last inserted ID is: " .
$last_id;
} else {
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}

mysqli_close($conn);

?>
```

MySQL and other languages

PHP – MySQLi Procedural: INSERT (prepared)

```
<?php
// prepare and bind
$stmt = mysqli_stmt_init($conn);
mysqli_stmt_prepare($stmt, "INSERT INTO MyGuests (firstname, lastname,
email) VALUES (?, ?, ?)");
mysqli_stmt_bind_param ($stmt, "sss", $firstname, $lastname, $email);

// set parameters and execute
$firstname = "John";
$lastname = "Doe";
$email = "john@example.com";
mysqli_stmt_execute ($stmt);

echo "New record created successfully";

mysqli_stmt_close($stmt);
mysqli_close($conn);
?>
```

MySQL and other languages

PHP – MySQLi Procedural: SELECT

```
<?php  
$sql = "SELECT id, firstname, lastname FROM MyGuests";  
$result = mysqli_query($conn, $sql);  
  
if (mysqli_num_rows($result) > 0) {  
    // output data of each row  
    while($row = mysqli_fetch_assoc($result)) {  
        echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " .  
$row["lastname"]. "<br>";  
    }  
} else {  
    echo "0 results";  
}  
  
mysqli_close($conn);  
?>
```

MySQL and other languages

PHP – MySQLi Object-Oriented: Connect, Query, Insert

```
<?php
$conn = new mysqli($servername, $username, $password);

$sql = "CREATE DATABASE myDB";
$conn->query($sql);

$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";

if ($conn->query($sql) === TRUE) {
    $last_id = $conn->insert_id;
    echo "New record created successfully. Last inserted ID is: " .
$last_id;
}
?>
```

MySQL and other languages

PHP – MySQLi Object-Oriented: Insert (prepared)

```
<?php
    // prepare and bind
$stmt = $conn->prepare("INSERT INTO MyGuests (firstname, lastname, email)
VALUES (?, ?, ?)");
$stmt->bind_param("sss", $firstname, $lastname, $email);

// set parameters and execute
$firstname = "John";
$lastname = "Doe";
$email = "john@example.com";
$stmt->execute();

echo "New records created successfully";

$stmt->close();
$conn->close();

?>
```

MySQL and other languages

PHP – MySQLi Object-Oriented: SELECT

```
<?php

$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        echo "id: " . $row["id"] . " - Name: " . $row["firstname"] . " " .
$row["lastname"]. "<br>";
    }
}
$conn->close();
?>
```

MySQL and other languages

Python (more info in BIT04-Scripting)

- `mysql.connector`

```
import mysql.connector

cnx = mysql.connector.connect(user='username', database='myDD')

cursor = cnx.cursor()

query = ("SELECT id, firstname, lastname FROM MyGuests")

cursor.execute(query)

cursor.close()

cnx.close()
```

- `MySQLdb`

```
import MySQLdb

db = MySQLdb.connect(host="localhost", user="username", passwd="password", db="myDB")

cur = db.cursor()

cur.execute("SELECT id, firstname, lastname FROM MyGuests")

db.close()
```