



CHALMERS
UNIVERSITY OF TECHNOLOGY



UNIVERSITY OF GOTHENBURG

Repetition and Exam Prep

Part 2 - SQL and MongoDB

Dr. Philipp Leitner



philipp.leitner@chalmers.se



@xLeitix



Goal for today

Provide you with detailed *learning objectives* for the second half of the course

“Study guide” for the exam

Repeat core materials and do some more exercises



General Structure of the Exam

4 hours - but you will probably not need the entire time

Combination of theory and practice

1/4 - 1/3 of the points will be theory-based

2/3 - 3/4 of the points will be based on applied tasks

No multiple-choice!

Closed book - only allowed material is what we hand out with the exam.

General Structure of the Exam

5 (top-level) questions

See exam stub on GUL

Areas:

Theory (multiple short subquestions)

EER

Relational Algebra

SQL

MongoDB

Not all areas are weighted the same

E.g., EER >> MongoDB



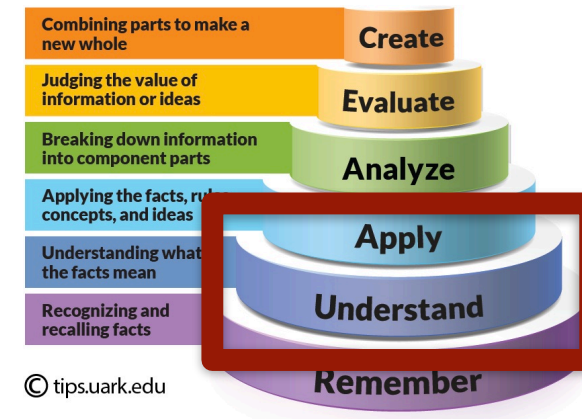
Repetition - Lecture 7

SQL DDL and Database Mapping

Summary of Central Learning Items

Understand and apply:

- **SQL DDL**
 - Creating and dropping databases
 - Inserting into and deleting from databases
 - Constraints
 - Data types
 - Handling foreign keys
- **Mapping EER to RM / SQL**
 - Concept of a cross-reference table
 - Mapping different types of inheritance



You'll need to know SQL syntax!

Types of Questions for this Block

Type 1 (this *may* be part of the exam):

“Here is an EER diagram, write SQL code to implement it.” *or*

“Here is an EER diagram, write down the relational model in RM notation.”

Type 2 (this *may* be part of the exam):

“Here is a relational model, write SQL code to implement it.”



Example for Type-2

EXECUTION(Exec_id, Job_id, Start, End)

JOB(Job_id, User)

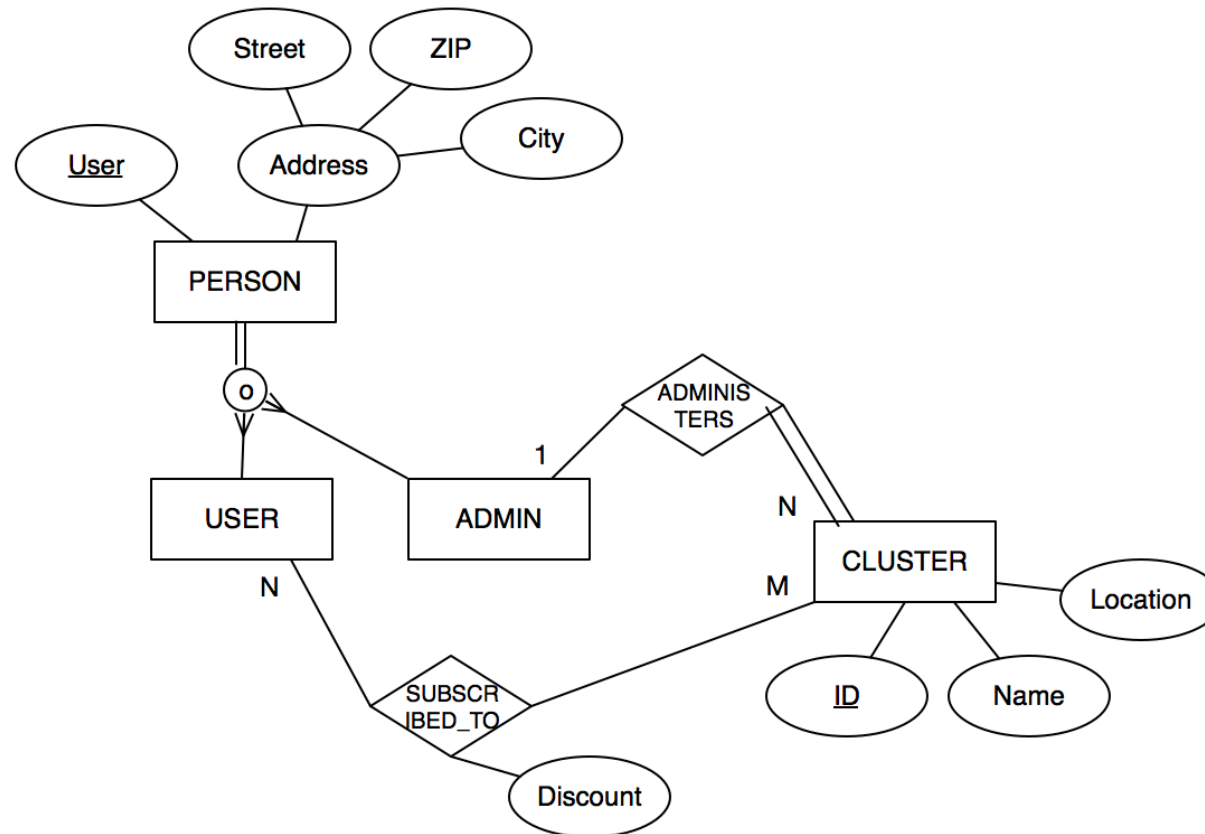
USER(Username, City, Street, ZIP)

Write SQL code to create the table EXECUTION. Assume that Job_id is a foreign key pointing to JOB. Make sure that the start date is not null and always before the end date.



Example for Type-1

Provide a relational model for the EER diagram on the next slide. If there are multiple ways to map the diagram, briefly document *why* you have chosen your approach. Use arrows from FK to PK to indicate foreign key relationships.





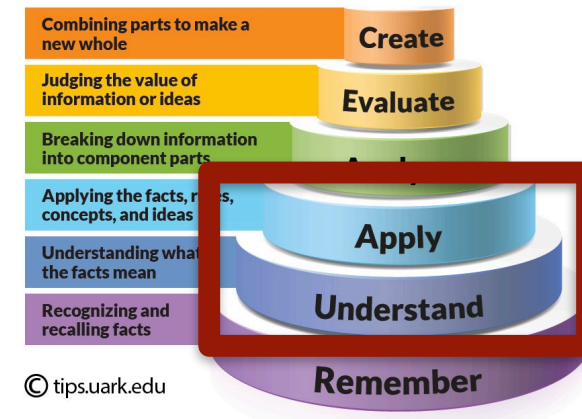
Repetition - Lecture 8

SQL Querying

Summary of Central Learning Items

Understand, explain, and apply:

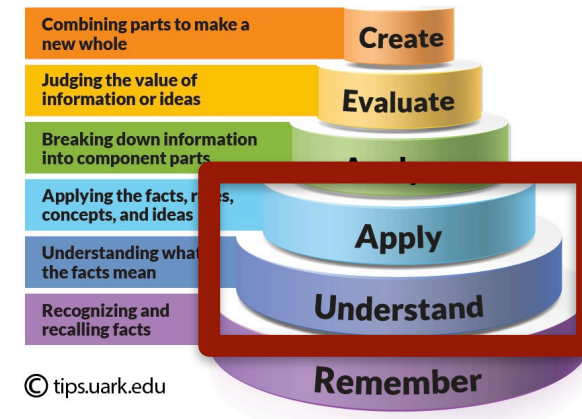
- Writing Queries in SQL (*need to know*)
 - SELECT, FROM, WHERE
 - Including joining and renaming
 - Condition operators and their syntax
 - Three-value logics
 - DISTINCT and sorting



Summary of Central Learning Items

Understand, explain, and apply:

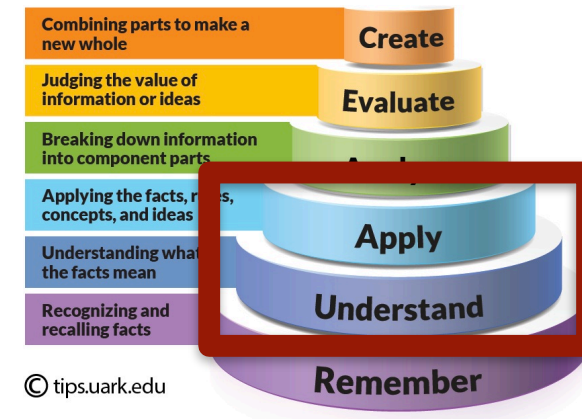
- Writing Queries in SQL (*still need to know*)
 - Aggregation
 - Grouping
 - Having



Summary of Central Learning Items

Understand, explain, and apply:

- Writing Queries in SQL (*good to know*)
 - OUTER JOINS
 - UNION, INTERSECT, EXCEPT
- Writing Queries in SQL (*nice to know*)
 - WITH
 - Subqueries



Types of Questions for this Block

Type 1 (this is going to be part of the exam **for sure**):

Given one or more relations or tables, formulate a small number of SQL queries.

Type 2 (this *may* be part of the exam):

Given one or more tables and table contents, write down the results of a small number of SQL queries *or* explain in your own words what a query would return if no state is given.

Example for Type-1

Consider the following Relational Model:

EXECUTION(Exec_id, Job_id, Start, End)

JOB(Job_id, User)

USER(Username, City, Street, ZIP)

Implement the following queries in SQL:

- Find the city of the user with the username 'philipp'.

Example for Type-1

Consider the following Relational Model:

EXECUTION(Exec_id, Job_id, Start, End)

JOB(Job_id, User)

USER(Username, City, Street, ZIP)

Implement the following queries in SQL:

- Find all executions of the user with the username 'philipp' (one result row per execution).

Example for Type-1

Consider the following Relational Model:

EXECUTION(Exec_id, Job_id, Start, End)

JOB(Job_id, User)

USER(Username, City, Street, ZIP)

Implement the following queries in SQL:

- Find all users who live in one of the following cities:
Gothenburg, Stockholm, Malmoe, Uppsala, Umea, or Lulea.



Example for Type-1

Consider the following Relational Model:

EXECUTION(Exec_id, Job_id, Start, End)

JOB(Job_id, User)

USER(Username, City, Street, ZIP)

Implement the following queries in SQL:

- Calculate the average execution duration of all executions for each user (the duration is the end time minus the start time). Take into account that some executions are still running (their end time is NULL), these should not be part of the calculation.

Example for Type-1

Consider the following Relational Model:

EXECUTION(Exec_id, Job_id, Start, End)

JOB(Job_id, User)

USER(Username, City, Street, ZIP)

Implement the following queries in SQL:

- Find all pairs of users that live in the same city. Return in the format (username1, username2, city). Make sure that your list does not contain duplicate entries.



Example for Type-2

Consider the following Relational Model:

EXECUTION(Exec_id, Job_id, Start, End)

JOB(Job_id, User)

USER(Username, City, Street, ZIP)

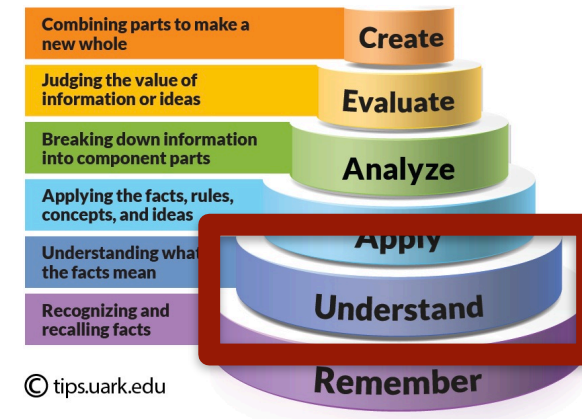
What does the following query return?

```
SELECT e1.Exec_id, e1.Job_id, (e1.End-e1.Start) as Duration
FROM EXECUTION e1
WHERE Duration >
    (SELECT AVG(e2.End-e2.Start) FROM EXECUTION e2);
```

Summary of Central Learning Items - JDBC

Understand and explain:

- **Embedded vs. interpreted vs. hidden SQL**
- **Basic structure of a JDBC program**
 - No need to write code, knowing basic steps is ok
- **Prepared statements**
- (JPA not part of exam material)





Repetition - Lecture 10

Transactions, Indices, Views

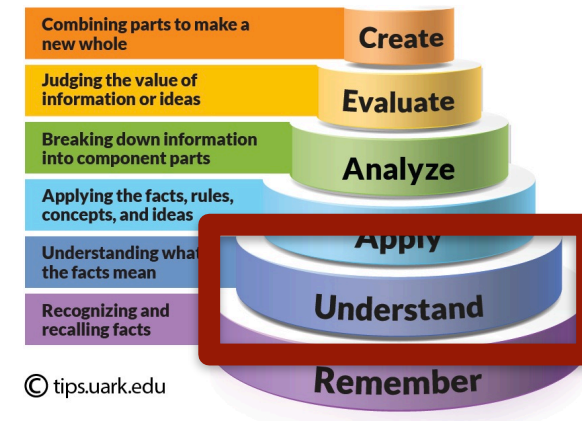
Summary of Central Learning Items - Transactions

Understand and explain: *(need to know)*

- What is a transaction, what is it good for?
- Transaction primitives
- ACID properties

Understand and explain: *(good to know)*

- Types of transactional violations
- Database isolation levels



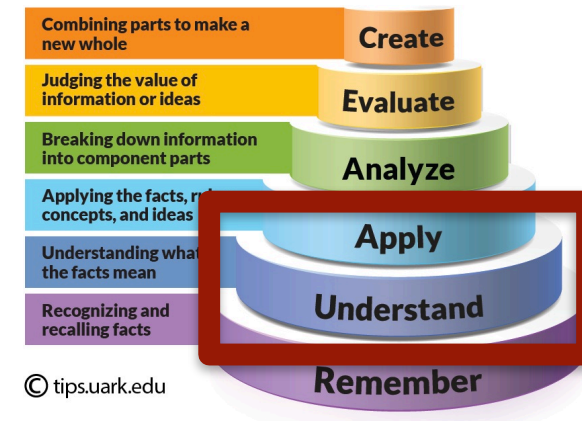
Example Question

Briefly describe the three transaction primitives. Come up with a sample interaction with the grid database that illustrates all primitives.

Summary of Central Learning Items - Views

Understand, explain, and apply:

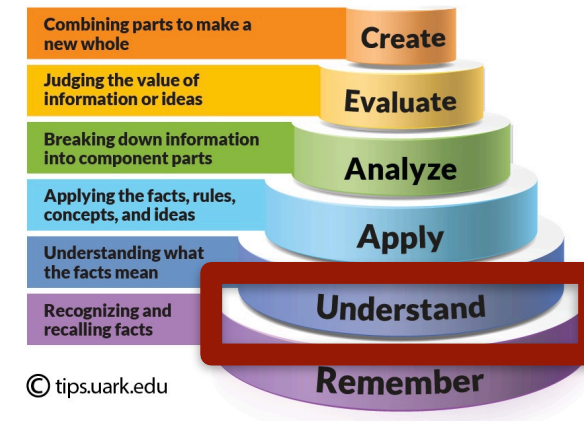
- Full table scans
- Query modification and materialized views
 - And SQL code to create them
- View materialization strategies



Summary of Central Learning Items - Indices

Understand and explain:

- General idea of indices
- Advantages and disadvantages





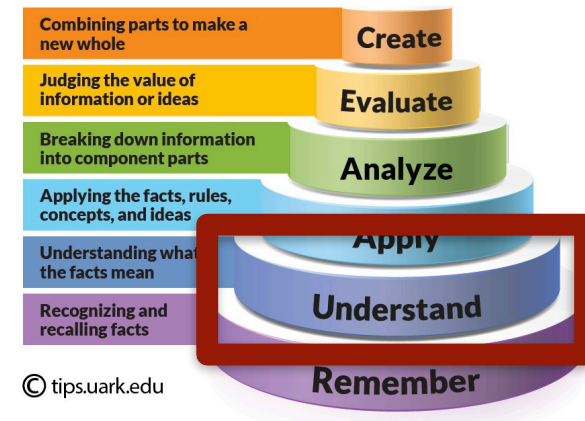
Repetition - Lecture 11 + 13

NoSQL and Map/Reduce

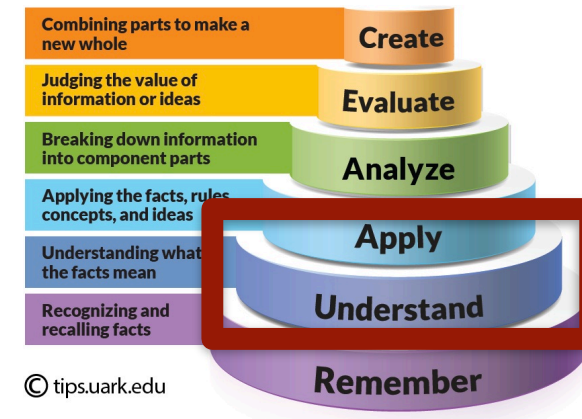
Summary of Central Learning Items - Distributed Databases

Understand and explain:

- Replication and sharding
- CAP theorem
- Eventual consistency



Summary of Central Learning Items - NoSQL



Understand and explain:

- Types of NoSQL databases (and their advantages)
- Basic idea of a document store

Apply:

- Analyse and understand MongoDB queries
 - (no need to write queries yourself)



Example

Describe what the following query does if executed in the MongoDB shell:

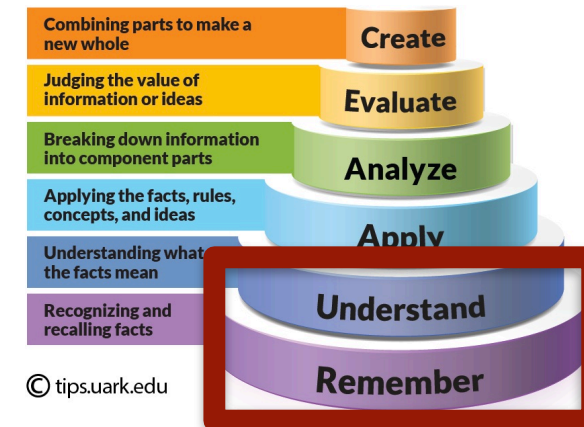
```
db.executions.find(  
  {"exec.job.user" : "philipp"})
```

Example

Describe what the following query does if executed in the MongoDB shell:

```
db.executions.mapReduce(  
  map, reduce,  
  {finalize: finalize, out: "mr_out"})
```


Summary of Central Learning Items - Big Data



Understand and explain:

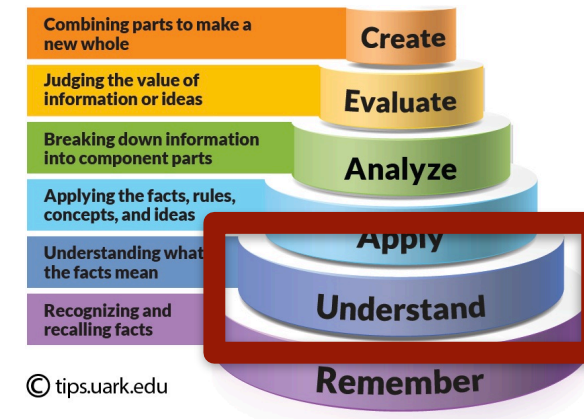
- The “5Vs” of Big Data
- Basic model of Map/Reduce



Repetition - Lecture 12

Representation

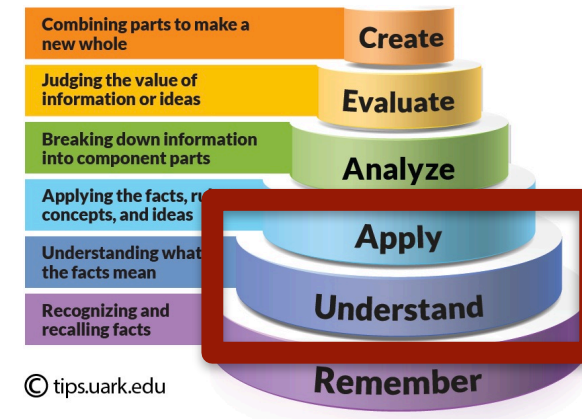
Summary of Central Learning Items - Representation



Understand and explain:

- Binary and text representations
- (Semi-)structured data
- XML, JSON, YAML

Summary of Central Learning Items - XML



Understand, explain, and apply:

- Hierarchical structure of XML
- Core well-formedness rules of XML
- Well-formed versus valid

Example

Is this minimal XML document valid? Why / why not?

```
<?xml version="1.0" encoding="UTF-8" ?>
<grids>
  <job jobid="1">
    <execution execid="2" start="23213">
  </job>
</execution>
</grids>
```

Example

Sketch the hierarchical tree structure of this minimal XML document. You can skip the XML header.

```
<?xml version="1.0" encoding="UTF-8" ?>
<grids>
  <job jobid="1">
    <execution execid="2" start="23213">
      bash my_runner.sh
    </execution>
  </job>
</grids>
```



Your Questions?

