CHALMERS
UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Repetition and Exam Prep

Part 1 - EER and Relational Algebra

Dr. Philipp Leitner

✉ philipp.leitner@chalmers.se

🐦 @xLeitix

Goal for today

Provide you with detailed *learning objectives* for the first 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

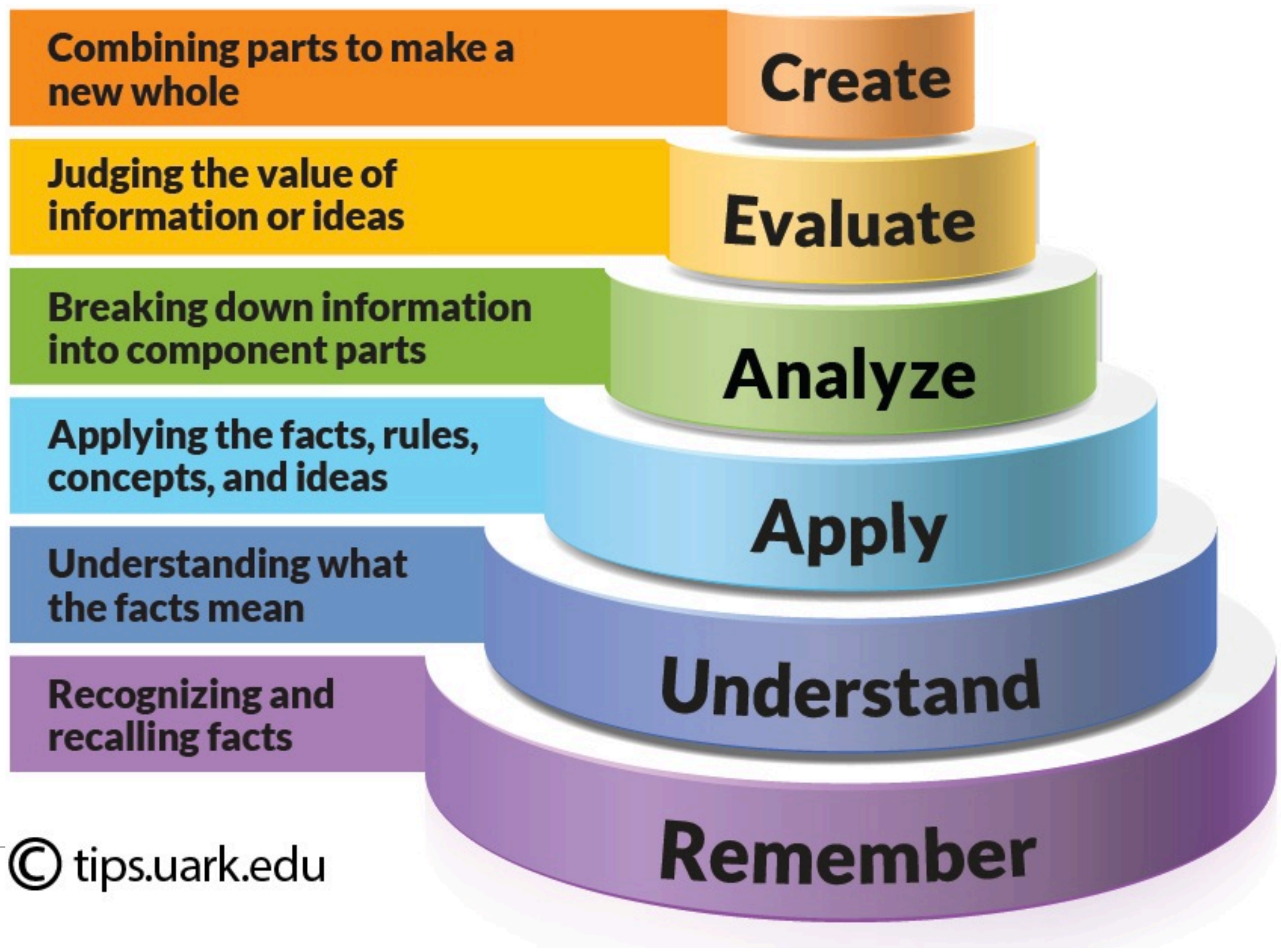
Combination of theory and practice

- 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.

Bloom Taxonomy



Combining parts to make a new whole — Create

Judging the value of information or ideas — Evaluate

Breaking down information into component parts — Analyze

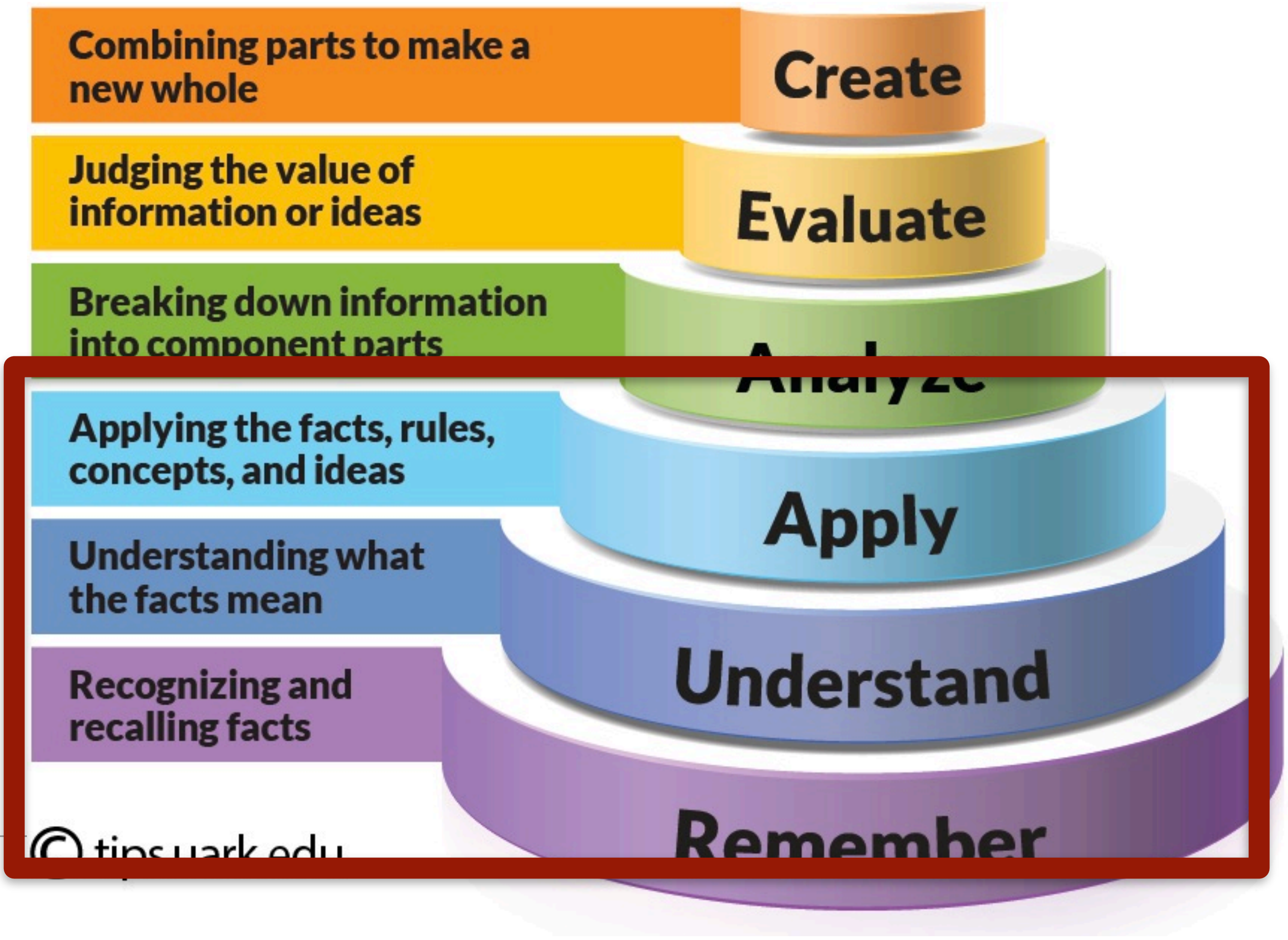
Applying the facts, rules, concepts, and ideas — Apply

Understanding what the facts mean — Understand

Recognizing and recalling facts — Remember

© tips.uark.edu

Bloom Taxonomy



Create: Combining parts to make a new whole

Evaluate: Judging the value of information or ideas

Analyze: Breaking down information into component parts

Apply: Applying the facts, rules, concepts, and ideas

Understand: Understanding what the facts mean

Remember: Recognizing and recalling facts

© tips.uark.edu

General Structure of the Exam

Revisited:

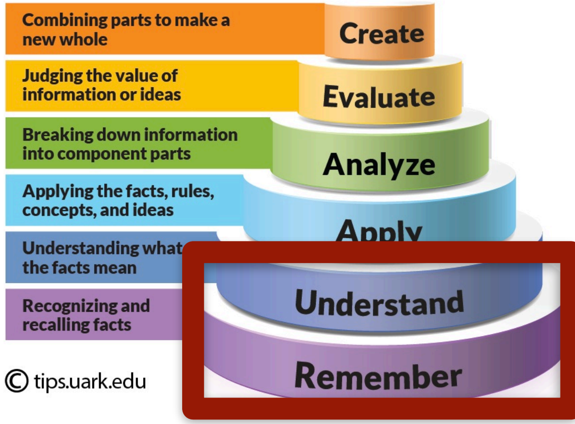
1/4 - 1/3 of the points for “Understand” type questions

2/3 - 3/4 of the points for “Apply” type questions

Repetition - Lecture 1

Introduction to Databases

Summary of Central Learning Items


Combining parts to make a new whole
Create
Judging the value of information or ideas
Evaluate
Breaking down information into component parts
Analyze
Applying the facts, rules, concepts, and ideas
Apply
Understanding what the facts mean
Understand
Recognizing and recalling facts
Remember
© tips.uark.edu

Understand and explain:

- **Data, information, and knowledge**
- **Basic database operations**
  CRUD - create, read, update, delete
- Databases as **collections of meta-data and data**
- **Advantages of using a DBMS**
- The ANSI/SPARC **Three-Schema Architecture**

Example Questions

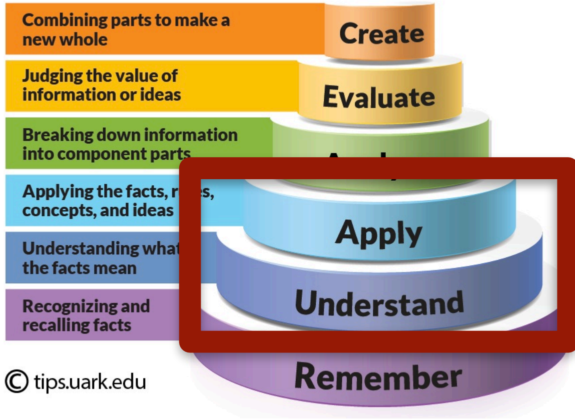
Draw the basic knowledge management pyramid. What’s the difference between data and information in this model?

Explain the concept of meta-data in your own words. Give an example of meta-data and data.

Repetition - Lecture 2 and 3

EER Block

Summary of Central Learning Items (1)

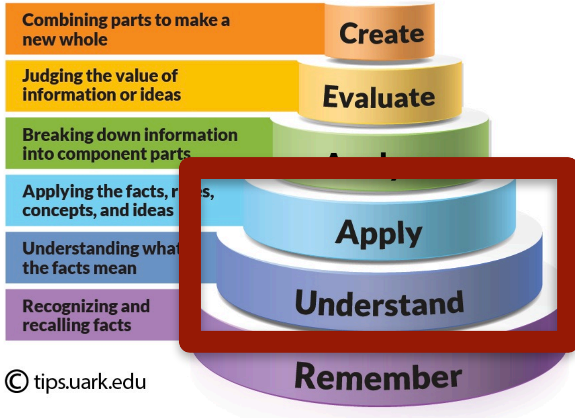

Combining parts to make a new whole
Create
Judging the value of information or ideas
Evaluate
Breaking down information into component parts
Applying the facts, r s, concepts, and ideas
Apply
Understanding what the facts mean
Understand
Recognizing and recalling facts
Remember

© tips.uark.edu

Understand and apply:

- **Basic ER concepts**
  - Entities, relationships, attributes, cardinalities, participation
  - Keys
- **Slightly more advanced concepts**
  - Composed attributes, multi-valued attributes
  - Weak entity types
  - Association attributes
  - Regular ("disjoint") generalizations

Summary of Central Learning Items (2)

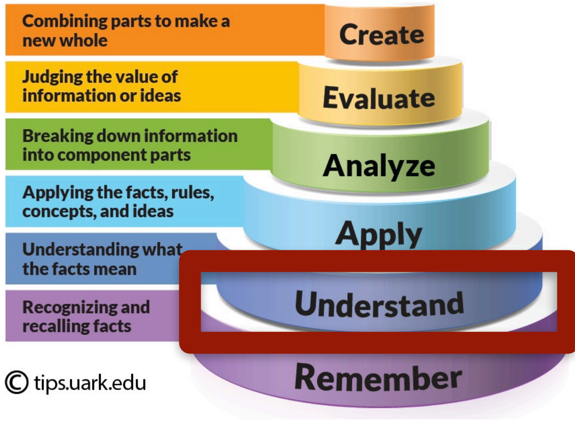

Combining parts to make a new whole
Create
Judging the value of information or ideas
Evaluate
Breaking down information into component parts
Applying the facts, r s, concepts, and ideas
Apply
Understanding what the facts mean
Understand
Recognizing and recalling facts
Remember

© tips.uark.edu

Understand and apply:

- **Advanced concepts**
  - Ternary relationships and their cardinalities
  - Overlapping generalizations
  - Derived attributes

Summary of Central Learning Items (3)


Combining parts to make a new whole
Create
Judging the value of information or ideas
Evaluate
Breaking down information into component parts
Analyze
Applying the facts, rules, concepts, and ideas
Apply
Understanding what the facts mean
Understand
Recognizing and recalling facts
Remember
© tips.uark.edu

Understand and explain:

- Difference between ER and EER
- Generalization versus specialization
- Multiple inheritance and union types

Types of Questions for this Block

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

"Here is a textual domain description, draw an EER diagram"

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

"Here is an EER diagram, answer some questions about it"

Type 3 (this *may* be part of the exam):

Short theory questions

E.g., "Shortly describe the difference between generalization and specialization."

Example for Type-1

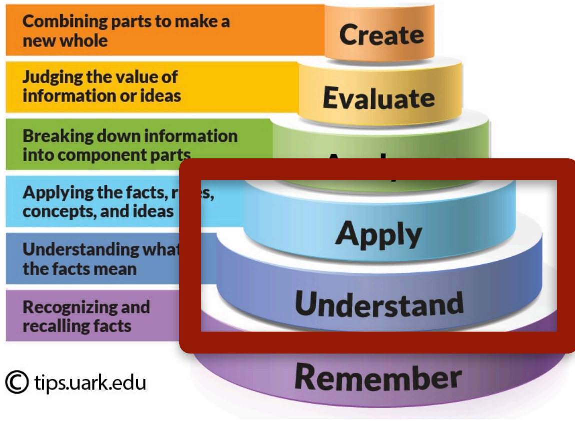
Consider the following outtake of the domain description of a high-performance computing system (“computational grid”). Model the described domain using an EER diagram, and using the notation we used in the course. *Use the 1,N,M notation for describing cardinalities rather than the min-max notation.*

If something is not specified, make an assumption and note it down in plain text.

- A computational grid consists of clusters of computers. Each cluster has an unique identifier (ID), a name, and a data center location. Every cluster contains a large (non-zero) number of computers, each with an IP address and hostname (both of which can be used to identify the computer). Every computer is in exactly one cluster. Computers have a defined amount of CPU and RAM.
- There are two types of persons in this system: users and admins. Admins can also be users, but they don’t have to. All persons have an unique user name and an address, which consists of street name, city, and postal code. Every admin administers multiple clusters, but each cluster has exactly one admin. Users, on the other hand, are subscribed to arbitrarily many clusters, and each cluster has many users. For each cluster subscription, a user may have been granted a custom discount.
- Users use the computational grid by scheduling jobs. Jobs are identified through a job ID and also contain a task to run. Every user can schedule as many jobs as they like, but every job belongs to exactly one user. Every job can be executed arbitrarily often, and runs on an arbitrary amount of computers. We need to store the start and end time of each execution. Executions are identified through an execution ID, which is only unique in the context of a job.

Repetition - Lecture 4 to 6

Relational Algebra Block


Combining parts to make a new whole
Create
Judging the value of information or ideas
Evaluate
Breaking down information into component parts
Applying the facts, r s, concepts, and ideas
Apply
Understanding what the facts mean
Understand
Recognizing and recalling facts
Remember

© tips.uark.edu

Summary of Central Learning Items - RM

Understand, explain, and apply:

- **Relational Modelling**
  - Relations, attributes, tuples, (valid and invalid) states
  - Different types of constraints
  - Primary keys, Candidate keys, super keys, foreign keys

Types of Questions for this Block

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

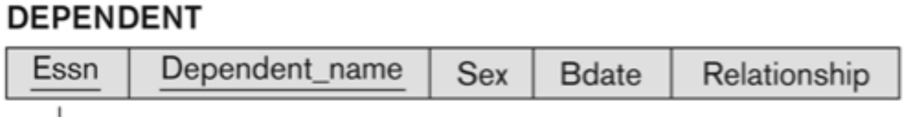
Short theory questions

E.g., “Can a valid relation state contain duplicate rows? Briefly explain your reasoning.”

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

Given some visualization (e.g., EER or something else that we used in the lecture) describe the relation using RM.

Example for Type-2

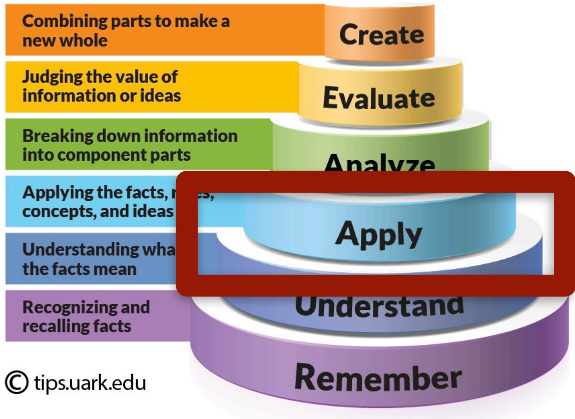


DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship

(1) Describe this relation using the relational model notation we used in the lecture. (2) What is the primary key of this relation? (3) Name one example of a superkey in this relation. Provide examples of a (4) valid and (5) invalid state.

Summary of Central Learning Items - RA


Combining parts to make a new whole
Create
Judging the value of information or ideas
Evaluate
Breaking down information into component parts
Analyze
Applying the facts, rules, concepts, and ideas
Apply
Understanding what the facts mean
Understand
Recognizing and recalling facts
Remember

© tips.uark.edu

Understand and apply:

- **Unary Relational Algebra Operators**
  - Select, Project, Assign, Rename
- **Relational Algebra Set Operators**
  - Union, Intersection, Difference, Cartesian Product
- **Join Operators**
  - Join, Equijoin, Natural Join, [Left|Right|Full] Outer Join
- **Aggregation and Grouping Operators**

- **Most important:** be able to combine operators in non-trivial expressions

Types of Questions for this Block

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

Given one or more relations (in mathematical or graphical notation), formulate a small number of RA queries.

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

Given one or more relations (in mathematical or graphical notation) and valid state(s), write down the results of a small number of RA expressions *or* explain in your own words what a expression 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)

Write the following RA expression:

- 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)

Write the following RA expression:

- How many jobs does the user with the username ‘philipp’ have?

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)

Write the following RA expression:

- Calculate the duration (end time minus begin time) of each execution. Create a new relation DURATION(Exec_id, Job_id, Duration) of the result. Assume that both, start and end time cannot be NULL.

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)

Write the following RA expression:

- Calculate the duration (end time minus begin time) of each execution. Create a new relation DURATION(Exec_id, Job_id, Duration) of the result. If the end time is NULL set the duration to NULL as well.

Examples for Type-2

Consider the following Relational Model:

EXECUTION(Exec_id, Job_id, Start, End)
JOB(Job_id, User)
USER(Username, City, Street, ZIP)
SUBSCRIBED_TO(Username, Cluster_id, Discount)
CLUSTER(Id, Name, Location)

What does the following RA expression return?

$$\mathscr{F}_{MAX\ Discount}\,(SUBSCRIBED\_TO)$$

Examples for Type-2

Consider the following Relational Model:

EXECUTION(Exec_id, Job_id, Start, End)
JOB(Job_id, User)
USER(Username, City, Street, ZIP)
SUBSCRIBED_TO(Username, Cluster_id, Discount)
CLUSTER(Id, Name, Location)

What's wrong with the following RA expression for returning the user details of the user with the highest discount?

$$DISCOUNT \leftarrow \mathscr{F}_{MAX\ Discount}\,(SUBSCRIBED\_TO)$$

$$DISCOUNT_{\bowtie(DISCOUNT.Discount=USER.Discount)}USER$$

Examples for Type-2

Consider the following Relational Model:

EXECUTION(Exec_id, Job_id, Start, End)
JOB(Job_id, User)
USER(Username, City, Street, ZIP)
SUBSCRIBED_TO(Username, Cluster_id, Discount)
CLUSTER(Id, Name, Location)

What does the following RA expression return?

$$DURATION \leftarrow \rho_{(Exec\_id, Job\_id, Duration)}(\pi_{Exec\_id, Job\_id, End-Start}(EXECUTION))$$

$$_{Job\_id}\mathscr{F}_{SUM\ Duration}\,(DURATION)$$

Examples for Type-2

Consider the following Relational Model:

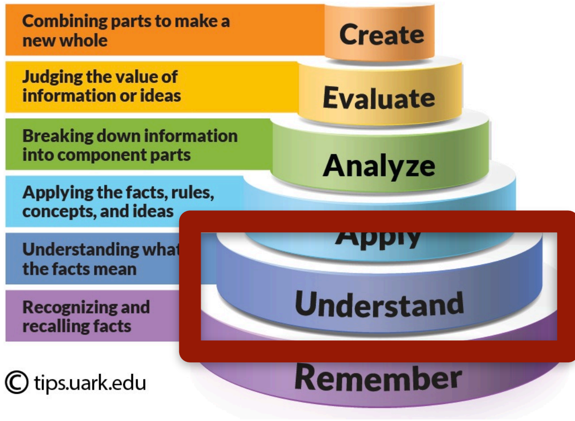
EXECUTION(Exec_id, Job_id, Start, End)
JOB(Job_id, User)
USER(Username, City, Street, ZIP)
SUBSCRIBED_TO(Username, Cluster_id, Discount)
CLUSTER(Id, Name, Location)

What does the following RA expression return?

$$S \leftarrow SUBSCRIBED\_TO$$

$$USER \bowtie_{(USER.Username=S.Username)} S \bowtie_{(S.Cluster\_id=CLUSTER.Id)} CLUSTER$$

Summary of Central Learning Items - Normalization


Combining parts to make a new whole
Create
Judging the value of information or ideas
Evaluate
Breaking down information into component parts
Analyze
Applying the facts, rules, concepts, and ideas
Apply
Understanding what the facts mean
Understand
Recognizing and recalling facts
Remember

© tips.uark.edu

<u>Understand and explain:</u>

- **"Informal" criteria for good relational design**
- **The problem with redundancy**
- **Functional dependencies**
  - Definition and being able to give examples
  - For good and bad FDs
- **Normalization**
  - Being able to define [1|2|3]NF

Example Question 1

Consider the following Relational Model:

EXECUTION(Exec_id, Job_id, Username)

JOB(Job_id, User)

USER(Username)

Give two examples of functional dependencies, one that is OK in 3NF and one that is not. Use the formal notation used in the lecture.

Example Question 2

Consider the following Relational Model:

EXECUTION(Exec_id, Job_id, Username)
JOB(Job_id, User)
USER(Username)

Is this model in 3NF? Briefly describe why / why not.

Your Questions?