



Introduction to Relational Database

Lucia D. Krisnawati

Overview

- Database & Database Management System
- Relational Database
- Simple SQL Queries
- Database normalization
- RDBMS for an Inverted Text Index

Database System Today

The screenshot shows the LMU Portal Start page with several sections highlighted by red circles:

- Lucia.Krisnawati** (User profile)
- Benutzerkonto** (User account)
- Mailbox: Posteingang** (Email inbox)
- LSF** (LSF-Stundenplan)
- LMU aktuell** (News section)
 - „Diversity“: Vielfalt an der LMU
 - „Unterschiede sind ein Gewinn“
 - Von den auf unterschiedliche Weise Besten die Besten gewinnen": Vizepräsidentin Barbara Conradt erklärt, was die LMU mit „Diversity“ verbindet: Gleichstellung und Inklusion, Familienfreundlichkeit und Chancengerechtigkeit.
- Studium**
 - [ZSB] Workshop "Lern- und Arbeitstechniken für studierende Eltern"
 - [ZSB] Einführungsveranstaltung "Studieren mit Kind" am 01.04.2014
 - [ZSB] Workshop "Studieren und Elternsein in Balance" am 27.03.2014 und 03.04.2014
 - [ZSB] Inforunde "Studieren mit Kind" am 25.03.2014 und 10.04.2014
 - [IA] Sprechstunde für Auslandstudienberatung enfällt
- Student und Arbeitsmarkt**
- Vorlesungszeit**
 - Noch 16 Tage
 - Semesterferien! Das Sommersemester beginnt am 07.04.2014. Rückmeldung zum SoSe bis zum 03.02.2014.
- LMU Wetter**
 - 7.2 °C
 - Weitere Wetterdaten
- Administratives**
 - Beitragskonto einsehen
 - Postanschrift ändern
 - Prüfungsverwaltung
- Links**
 - Online-Selbstbedienungsfunktionen
 - Beratungsstellen
 - Prüfungsverwaltung
 - Verwaltung A-Z
 - Personalrat
 - Schadensmeldung / Störmeldung
- News Fachschaft Psychologie**
- News der Universitätsbibliothek**
 - Ausweitung des Campuslieferdienstes
 - Zentralbibliothek: Schließung PC-Raum 2 vom 28.-31. März
 - Zentralbibliothek: Serviceschalter schließt ab dem 24. März bereits um 18 Uhr
 - Medizinstudium: Lernprogramm „AMBOSS“ ersetzt „Examen online Klinik“
 - Fachbibliothek Medizinische Lesehalle: Schließung vom 15.-19. April

Database System Today

Google LMU gebäude|München

Get directions My places

LMU gebäude, München

A Ludwig Maximilian University of Munich
Geschwister-Scholl-Platz 1, 80539 München
089 21800 · uni-muenchen.de
4.4 ★★★★★ 19 reviews
"Mit LMUInnovativ hat die Universität **München** 2004 einen Prozess eingeleitet, durch den die Profilbildung und -anpassung im Bereich Forschung gewährleistet wird." -

B Klinikum der Universität München
Nußbaumstraße 20, 80336 München
089 51600 · klinikum.uni-muenchen.de
4.1 ★★★★★ 9 reviews
"Das Krankenhaus selber ist allerdings ein sehr altes Gebäude (Duschen und Toiletten auf dem Gang; 4-6 Personen Zimmer)." - medmonitor.de

C Universitätsbibliothek der LMU München – Fachbibliothek Biozentrum
Großhaderner Straße 4, 82152 München
089 218074004 · biozentrum.ub.uni-muenchen.de
"Homepage der Universitätsbibliothek der **LMU München**. Hier finden Sie Bücher, E-Books, Datenbanken...Medien aller Art." - uni-muenchen.de

D Cafeteria TU Stammgelände
Arcisstraße 21, 80333 München
089 281003 · studentenwerk-muenchen.de
1 review
"Auf dieser Karte sehen Sie die Menschen, Mensarien, StuCafés und Espresso-Bars des Studentenwerks in **München**. Bitte beachten Sie die Öffnungs- und ..." - studentenwerk-muenchen.de

E Studentenwerk München
Leopoldstraße 15, 80802 München
089 38196 ext. 0 · studentenwerk-muenchen.de
"Das Referat für Bildung und Sport der Stadt **München** veranstaltet in Kooperation mit der Agentur für Arbeit, der Hochschule **München**, der **LMU**, der TU ..." - studentenwerk-muenchen.de

Indersdorf Karpfhofen Röhrmoos Fahrenhausen Haimhausen Unterschleißheim Eching Neufahrn bei Freising Dietersheim Dachau Holzgarten Oberschleißheim Mallersthofer Holz mit Heiden Hochbrück Ismaning Moosach Moosach Pasing-Obermenzing Allach-Untermenzing Gröbenzell Neuhausen-Nymphenburg Schwabing-Freimann Eichenau Puchheim Unterpfaffenhofen Hardheim Unterbrunn Gauting Stockdorf Krailling Planegg Neuried Thalkirchen Untergiesing-Harlaching Thalkirchen-Obersendling-Forstenried-Fürstenried-Solln Geisenbrunn Unterbrunn Starnberg Buchendorf Buchenhain Frundsbergerhöhe Deisenhofen Brunnthal Höhenkirchen-Siegristsbrunn Starnberg Kempfenhausen Schäftlarn Fründsbergerhöhe Deisenhofen Brunnthal Höhenkirchen-Siegristsbrunn Egling Moosach Bruck Glonn

Database System Today

The image displays the sign-up pages for three popular social media platforms:

- Facebook Sign Up:** Shows a mobile phone icon with a green dotted line leading to it, indicating mobile connectivity. Fields for First Name, Last Name, Your Email, and Re-enter Email are present.
- Twitter Welcome:** Shows two smartphones displaying Twitter feeds. Buttons for "Download on the App Store" and "ANDROID APP ON Google play" are visible.
- WhatsApp Sign Up:** Shows a smartphone displaying a messaging interface with a green speech bubble icon. A "Download WhatsApp" button is prominent.

- Tremendously huge data processing
- Horizontal Scalability
- Concurrency Model

The image shows the official WhatsApp website homepage, featuring the following elements:

- Header:** The WhatsApp logo (green speech bubble) and navigation links: Home, Download, FAQ, Blog, Contact, and a language selection dropdown set to English.
- Slogan:** "Simple. Personal. Real Time Messaging."
- Call-to-Action:** A large "Download WhatsApp" button.
- Smartphone Preview:** A central smartphone displaying a messaging conversation between Alice Whitman and Francisco Whitman.
- Features:** Three small images illustrating Group Chats, Send photos & videos, and Share locations.
- Device Support:** Icons for various mobile devices (iPhone, BlackBerry, Nokia, etc.) indicating platform availability.

What are DB & DBMS than?

- A database (DB) is a collection of data describing the activities of 1 or more related organization, eg. University database:
 - Entities: students, faculty, courses, classrooms
 - Relationship between entities:
 - Students' enrollment in courses
 - Faculty teaching courses
 - The use of rooms for courses
- A Database Management System (DBMS) is a software designed to assist in maintaining & utilizing large collection of data eg.:
 - Part of software industry: Oracle, Microsoft, Sybase
 - Open source:
 - Relational: MySQL, PostgreSQL, SQLite
 - Text search: APACHE Lucene (SOLR, HADOOP), Ferret,

Storing Data: File System vs DBMS

- Data can be stored in RAM
 - That is what most programming language offers
 - RAM is fast, random access but volatile
- File System offered by every OS:
 - Stores data in files with diverse formats in disk
 - Implication \Rightarrow program using these files depend on the knowledge about that format
 - Allows data manipulation (open, read, write, etc.)
 - Allows protection to be set on a file
 - Drawbacks:
 - No standards of format
 - Data duplication & dependence
 - No provision for concurrency & security

Quizzes

- Quiz 1:
 - You & your colleague are editing the same file.
 - You both save it at the same time
 - Whose changes survive?
- Quiz 2:
 - You & your colleagues login in the LMU portal.
 - Both of you are editing your addresses.
 - You both click the send button at the same time
 - Whose changes survive?

Storing Data: File System vs DBMS

- Database Management system:
 - Simple, efficient, ad hoc queries
 - Concurrency controls
 - Recovery, Benefits of good data modelling
 - Stores information in disks
 - This has implication for database design:
 - READ : transfer data from disk to main memory (RAM)
 - WRITE : transfer data from RAM to disk
 - In relational DBMS:
 - Information is stored as *tuples* or *records* in *relations* or *tables*.
 - Making use of relational Algebra

Relational Database

- Relational Database Management System (RDBMS) consists of:
 - A set of tables
 - A schema
- A schema:
 - is a description of data in terms of data model
 - Defines tables and their attributes (field or column)
- The central data description construct is a relation:
 - Can be thought as records
 - eg. information on student is stored in a relation with the following schema:

Student(***sid***: string, ***sname***: string, ***login***: string, ***gpa***: numeric)

Relational Database

- Tables \equiv relation:
 - is a subset of the Cartesian product of the domains of the column data type.
 - Stores information about an entity or theme
 - Consist of columns (fields) and rows (records).
 - Rows \equiv tuple, describing information about a single item, eg. A specific student
 - columns \equiv attributes, describing a single characteristic (attributes) of its item, eg. Its ID number, GPA, etc
 - Every row is unique & identified by a key
- Entity is
 - an object in the real world that is distinguishable from other objects. eg. Students, lecturers, courses, rooms.
 - Described using a set of attributes whose domain values must be identified.
 - The attribute 'name of Student' \Rightarrow 20-character strings

Creating Relational Database

- How to create relational database?
 - Need RDBMS (MySQL, Oracle, etc)
 - Just take MySQL as an open source RDBMS
 - With user Inteface
 - eg. phpMyAdmin → providing graphical user interface
 - Free to use any scripts or programming languages
 - Using SQL commands in terminal
 - Using SQL integrated in your code

Creating Relational Database

- How to create relational database in GUI?
 - Step 1: install XAMPP (just an example)
a cross-platform Apache HTTP Server, MySQL Server & interpreters for script
 - Step 2: start your XAMPP first:
`/xampp_or_lampp_path start`
eg. `/opt/lampp/lampp start`
 - Open your browser, and type:
`localhost/phpmyadmin`

RDBMS Example

- Database Server: MySQL 5.5.27
- Web Server: Apache through XAMPP Package

The screenshot shows the 'Databases' section of the phpMyAdmin interface. At the top, there is a navigation bar with tabs: Databases, SQL, Status, Users, Export, Import, and Settings. Below the navigation bar, the title 'Databases' is displayed. A red circle highlights the 'Create database' button and its input field. The input field contains the text 'IR14'. To the right of the input field is a dropdown menu labeled 'Collation' and a 'Create' button. Below this, a table lists existing databases: classification, information_schema, mysql, phpmyadmin, and test. Each database entry includes a 'Check Privileges' link. At the bottom of the list, it says 'Total: 5'. At the very bottom of the page, there are links for 'Check All / Uncheck All With selected:', 'Drop', and 'Enable Statistics'.

Database	Check Privileges
classification	Check Privileges
information_schema	Check Privileges
mysql	Check Privileges
phpmyadmin	Check Privileges
test	Check Privileges

Total: 5

Check All / Uncheck All With selected: Drop

Enable Statistics

RDBMS Example

- Creating table, defining attributes & their domains

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A.I.	Comment
	INT		None						
	INT		None						
	INT		None						
	INT		None						

Table comments:

Storage Engine: **InnoDB**

Collation:

PARTITION definition:

RDBMS Example

- Creating table, defining attributes & their domains

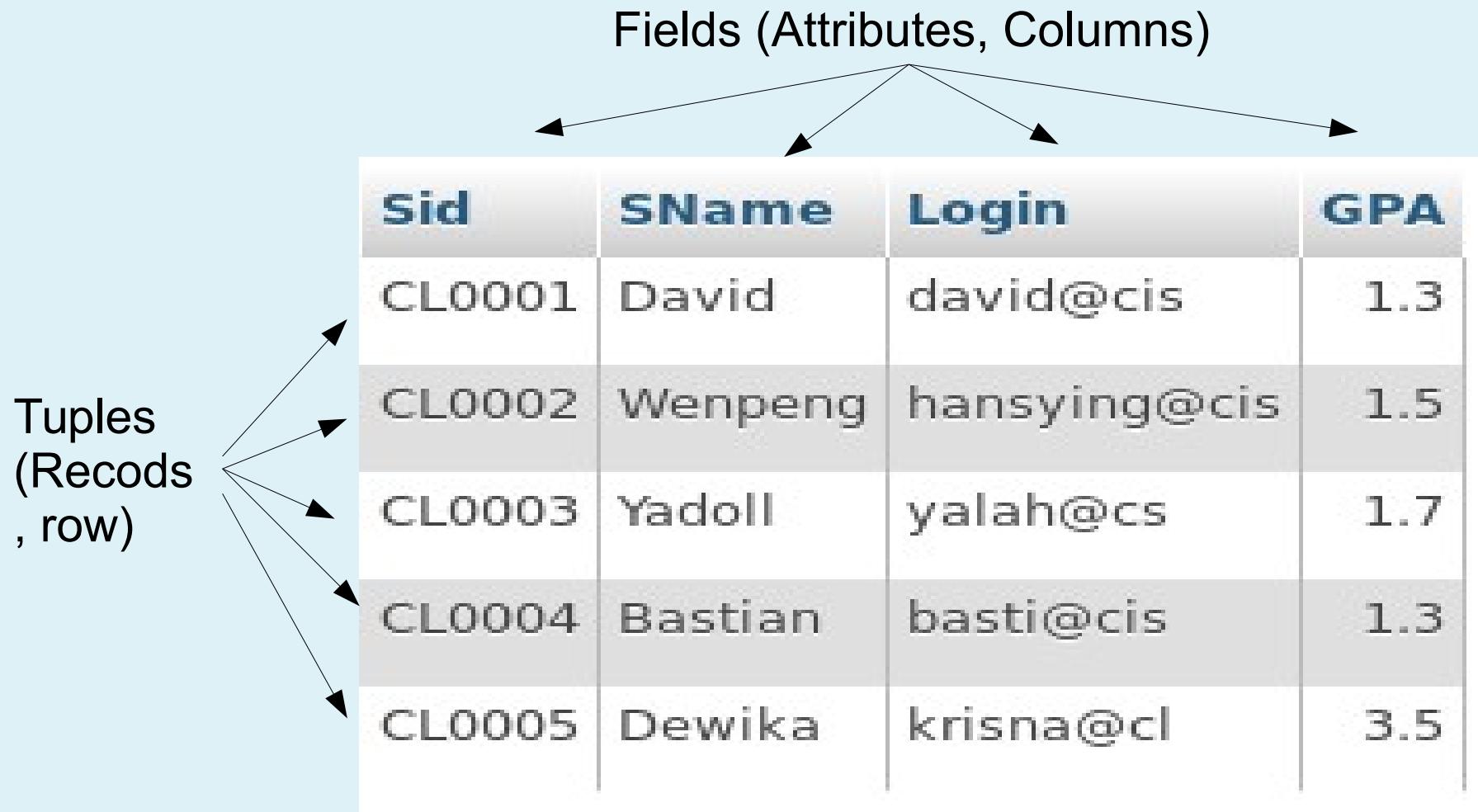
The screenshot shows the MySQL Workbench interface with the 'Structure' tab selected. The table 'IR14' is displayed with the following columns:

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	Sid	varchar(6)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index ▾ More
2	SName	varchar(35)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index ▾ More
3	Login	varchar(25)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index ▾ More
4	GPA	float(2,1)			No	None		Change Drop Browse distinct values Primary Unique Index ▾ More

At the bottom, there are buttons for 'Print view', 'Relation view', 'Propose table structure', 'Track table', 'Add 1 column(s)', 'At End of Table', 'At Beginning of Table', 'After Sid', and 'Go'. There is also a '+ Indexes' button and an 'Information' button.

RDBMS Example

- Each relation is defined to be a set of unique tuples of rows



Key Constraints

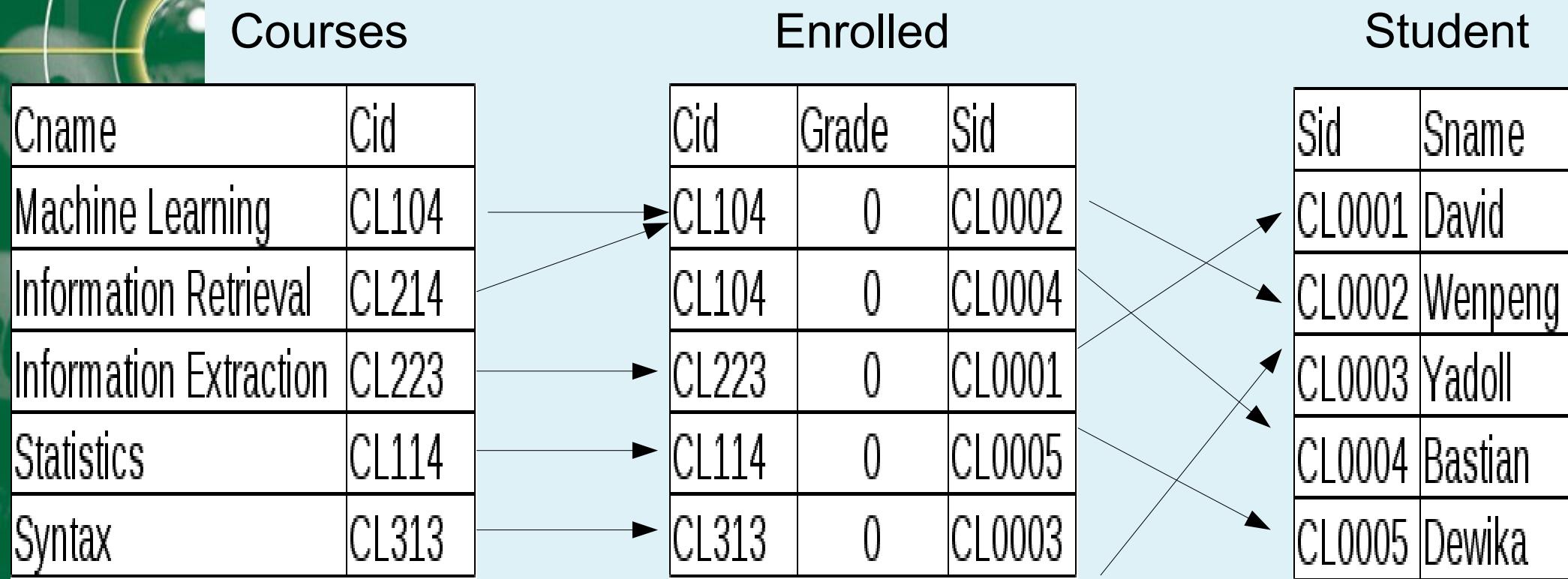
- Key constraint is a statement that a certain minimal subset of the relation is a unique identifier for a tuple.
- Two Types of keys:
 - Primary key:
 - Foreign key
- Primary key:
 - a unique identifier for a tuple (row)
 - Sid is a primary key for student,
 - Cid is a primary key for Course
 - Primary key fields are indexed

Key Constraints

- Foreign key:
 - A kind of a logical pointer
 - a key to refer to relation with other tables & should match the primary key of the referenced relation
 - Foreign key fields are also often indexed if they are important for retrieval.
- courses(Cid, Cname, Instructor, Semester)
Student(Sid, Sname, login, GPA)
- How do you express which students take which course?

Key Constraints

- Need a new table :
 - enrolled(Cid, grade, Sid)
 - Sid/Cid in enrolled are foreign keys referring to Sid in Student table & Cid in Courses.



Relations

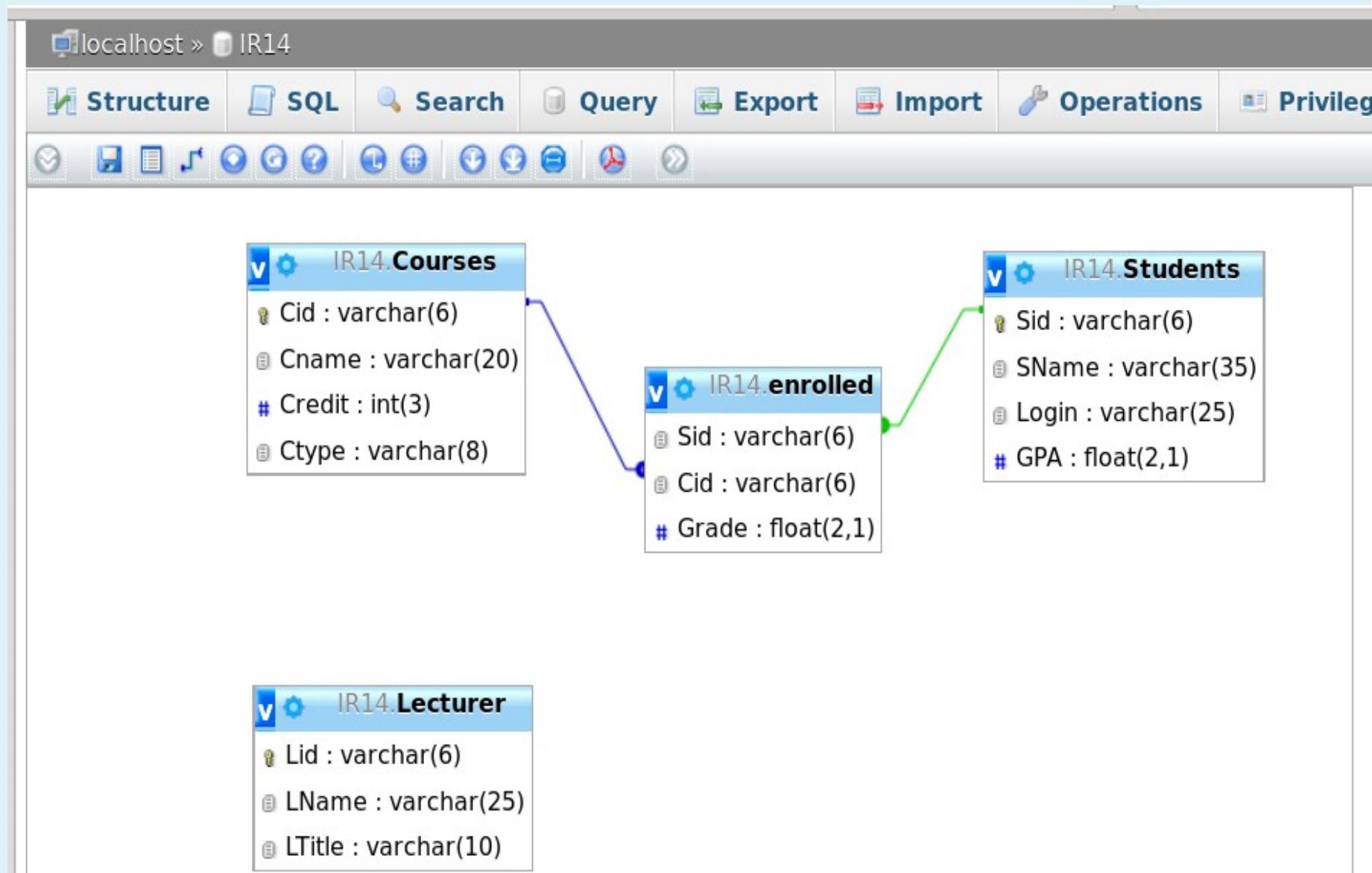
- One to one :
 - Each primary key relates only one record in related table
- One to many:
 - The primary key relates to one or many records in related table
- Many to Many:
 - The primary key relates to many records in related table, and a record in related table can relate to many primary keys on another table



Storing Relationships using Keys

- Modeling data is one thing, storing it in a database is another one.
- In relational database, the 'rules' are:
 - If the relationship to be stored is 1:N, place the attribute identified as the primary key from the one table as a foreign key in another table.
 - If the relationship to be stored is M:N, a new table structure must be created to hold the association. This 'bridge' table will have as foreign key attributes, the primary key of each table that is part of relationship
 - The key for the 'bridge' table then becomes either:
 - The combination of all the foreign keys OR
 - A new attribute will be added as a surrogate key

Storing Relationships using Keys



Indexes in MySQL

- A database index is
 - a data structure that improves the speed of operations in a table
 - Unseen table created by DB engine that keeps indexed fields and its pointers to each record into the actual table.
- Indexes in MySQL:
 - Primary key
 - Unique indexes:
 - All values in the indexed column must be distinct though it's unnecessarily indexed as a primary key
 - Index:
 - Refers to a non-unique index, used for speeding the retrieval

Indexes in MySQL

- Indexes in MySQL:
 - Fulltext:
 - An index created for full text searches
 - Supporting storage engines: InnoDB & MyISAM
 - Data type: CHAR, VARCHAR, TEXT
 - Spatial Index:
 - for spatial data types
 - Uses R-tree indexes
- Example of index usage:
 - „Find all students with GPA < 1.7“
 - May need to scan the entire table
 - Index consists of a set of entries pointing to locations of each search key

Data Type in MySql

- String:
 - Char, varchar, text, (tiny, medium, long)
 - Binary, varbinary
 - Blob (tiny, medium, long), enum, set
- Date & time
- Numeric
 - Int (tiny, small, medium, big)
 - Decimal, float, double, real
 - BIT, boolean, serial
- Spatial:
 - Geometry, point, linestring, polygon, etc

SQL

- Structured Query Language (SQL):
 - Is a standard language used to communicate with a relational database.
 - Is used in conjunction with procedural or object-oriented languages/scripts such as Java, Perl, Ruby, Python, etc
- Sql basic conventions:
 - Each statement begins with a command, eg. CREATE, SELECT
 - Each statement ends with delimiter usually a semicolon (;
 - Statements are written in a free-form style, eg. SELECT...FROM... WHERE...
 - SQL statement is not case-sensitive, except inside string constant, eg SELECT...FROM... WHERE SName = 'Yadolf'²⁷

Simple SQL Queries

- The basic form of SQL Queries is:

SELECT select-list (column_name)

FROM from-list (table_name)

WHERE condition

- Selecting all students with GPA above 1.7

SELECT Sid, Sname FROM student WHERE GPA <= 1.7

- Selecting all information from a table

SELECT * FROM enrolled

- Selecting course name with pattern matching

SELECT Cname FROM Courses WHERE Cname LIKE 'Machine %'

Simple SQL Queries

- INSERT:

```
INSERT INTO `Students` VALUES (CL0001, David, david@cis,  
1,3 )
```

```
INSERT INTO `Students` VALUES (sid, sname, login, gpa )
```

- ALTER:

```
ALTER TABLE `Students` ADD `Intakeyear`
```

```
ALTER TABLE `Lecturer` ADD INDEX(`courses`)
```

- Using logical connectives:

- AND, OR, NOT may be used to construct a condition

```
SELECT `cname` FROM `courses` WHERE semester =  
'summer' AND ctype = 'seminar'
```

- Joining Tables:

- SELECT `Sname` FROM `Students`, `Courses` WHERE
Students.sid = Courses.sid

Simple SQL Queries

- Creating Table:

```
CREATE TABLE `Students` (
    `Sid` varchar(6) NOT NULL,
    `SName` varchar(35) NOT NULL,
    `Login` varchar(25) NOT NULL,
    `GPA` float(2,1) NOT NULL,
    PRIMARY KEY (`Sid`)
) ENGINE=InnoDB CHARSET= Latin1;
```

Creating Database Through Terminal

- Open your terminal console
- Go to the path where you save your MySql
- If you install XAMPP :
 - You need to start XAMPP as a SU/root
 - to get the action commands (in Linux), type:
`/opt/lampp/lampp`
 - Start only MySQL Server, type:
`/opt/lampp/lampp startmysql`
 - To stop MySQL, type:
`/opt/lampp/lampp stopmysql`
 - To start XAMPP (Apache, MySQL & others), type:
`/opt/lampp/lampp start`

Creating Database Through Terminal

- If you install XAMPP :
 - go to the path where mysql is saved, in Linux it is usually saved in bin, so type:
`/opt/lampp/bin/mysql -uusername -ppassword`
 - If you are already in mysql path:
 - To see the databases. Type:
`SHOW DATABASES ;`
 - To create a database, use SQL command:
`CREATE DATABASE database_name ;`
 - Creating database does not select it for use, so type:
`USE database_name ;`
 - To delete database:
`DROP DATABASE database_name ;`
 - Use SQL commands to create tables, do table operation, etc

Creating Database Through Terminal

```
+-----+  
1 row in set (0.00 sec)  
  
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| IR14 |  
| cdccl |  
| classification |  
| mysql |  
| performance_schema |  
| phpmyadmin |  
| test |  
+-----+  
8 rows in set (0.00 sec)  
  
mysql> create database information_retrieval  
      -> ;  
Query OK, 1 row affected (0.00 sec)  
  
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| IR14 |  
| cdccl |  
| classification |  
| information_retrieval |  
| mysql |  
| performance_schema |  
| phpmyadmin |  
| test |  
+-----+  
9 rows in set (0.00 sec)
```

Database Normalization

- Normalization:
 - is the process of evaluating & correcting the structures of the tables in a database
 - The goal:
 - to minimize or remove data redundancy
 - To optimalize the data structure
 - Accomplished by thoroughly investigating the various data type and their relationships with one another.
- Data redundancy:
 - The repeat of key fields usages in other tables

Database Normalization

- Functional dependencies:
 - Require that the value for a certain set of attributes determines uniquely the value for another set of attributes
 - are akin to a generalization of the notion of a key
 - Let R be a relation and

$$\alpha \subseteq R \text{ and } \beta \subseteq R$$

The functional dependency :

$$\alpha \rightarrow \beta$$

holds on R and only if for any tuples t_1 & t_2 that agree on the attributes α , they also agree on the attributes β .

- That is, $t_1[\alpha] = t_2[\alpha] \rightarrow t_1[\beta] = t_2[\beta]$

Database Normalization

- Functional dependencies

Example: consider student(Sid, Sname, DeptId)
instance of student.

<u>Sid</u>	<u>Sname</u>	<u>DeptId</u>
CL12001	JOHN	13
CL13050	WENPENG	13
DE10003	ALDI	15
PS11123	ILJA	11
IT09256	LISANDRO	09
CL13075	MATTHEW	13

Is this true?	Yes	No
$\text{Sid} \rightarrow \text{Sname}$		
$\text{Sid} \rightarrow \text{DeptId}$		
$\text{Sname} \rightarrow \text{DeptId}$		
$\text{Sname} \rightarrow \text{Sid}$		
$\text{DeptId} \rightarrow \text{Sname}$		
$\text{DeptId} \rightarrow \text{Sid}$		

Database Normalization

- Functional dependencies

Example: consider student(Sid, Sname, DeptId)
instance of student.

<u>Sid</u>	<u>Sname</u>	<u>DeptId</u>
CL12001	JOHN	13
CL13050	WENPENG	13
DE10003	ALDI	15
PS11123	ILJA	11
IT09256	LISANDRO	09
CL13075	MATTHEW	13

Is this true?	Yes	No
$\text{Sid} \rightarrow \text{Sname}$	✓	
$\text{Sid} \rightarrow \text{DeptId}$	✓	
$\text{Sname} \rightarrow \text{DeptId}$		✓
$\text{Sname} \rightarrow \text{Sid}$		✓
$\text{DeptId} \rightarrow \text{Sname}$		✓
$\text{DeptId} \rightarrow \text{Sid}$		✓

Database Normalization

- examine the following poor database design:

	← T →	▼	Sid	Cname	time	room	Lid
	Edit		CL0001	Machine Learning	Wed 10.15	L155	PR145
	Edit		CL0002	Information Retrieval	Tue 12.15	C131	PD220
	Edit		CL0003	Machine Learning	Wed 10.15	L155	PR145
	Edit		CL0004	Information Extraction	Thu 10.00	C149	PR111

- Problems:
 - No need to repeatedly store the class time & Professor ID
 - Which one is the key?

Database Normalization

- First Normal Form (1NF):
 - A row of data cannot contain a repeating group of data.
 - Each row of data must have a unique identifier, i.e primary key
- This can be done by
 - Eliminating the repeated groups of data through creating separate tables of related data
 - Identify each set of related data with a primary key
 - All attributes are single valued (1 data type) & non-repeating

- Student information:

<i>Sid</i>	<i>Sname</i>	<i>Major</i>	<i>Minor</i>	<i>IntakeYear</i>
------------	--------------	--------------	--------------	-------------------

- Course information

<i>Cid</i>	<i>Cname</i>	<i>Lid</i>	<i>Time</i>	<i>Room</i>
------------	--------------	------------	-------------	-------------

- Lecturer Information

<i>Lid</i>	<i>Lname</i>	<i>Ltitle</i>
------------	--------------	---------------

Database Normalization

- Second Normal form (2NF):
 - A table should meet 1NF
 - There must not be any partial dependency of any column on primary key (Records should not depend on anything other than a table's primary key)
- Recall our poor database design:
 $\text{Sid} \rightarrow \text{Cname}$ or $\text{Cname} \rightarrow \text{time}$?

	← T →	▼	Sid	Cname	time	room	Lid
<input type="checkbox"/>	Edit Copy Delete	CL0001	Machine Learning	Wed 10.15	L155	PR145	
<input type="checkbox"/>	Edit Copy Delete	CL0002	Information Retrieval	Tue 12.15	C131	PD220	
<input type="checkbox"/>	Edit Copy Delete	CL0003	Machine Learning	Wed 10.15	L155	PR145	
<input type="checkbox"/>	Edit Copy Delete	CL0004	Information Extraction	Thu 10.00	C149	PR111	

Database Normalization

- Second Normal Form (2NF) solution:
 - **Create** separate tables for sets of values that apply to multiple records
 - **Relates** the tables with a **foreign key**
 - **Remove** subsets of data that apply to multiple rows of a table and **place** them in separate tables
- enrolled
 - | | | |
|-----|-----|-----------|
| Sid | Cid | grade (?) |
|-----|-----|-----------|
 - What do we do with the attribute time, room, & Lid?

Database Normalization

- Third Normal Form (3NF):
 - Eliminate all attributes (columns) that do not directly dependent upon the primary key
 - Each non-primary key attribute must be dependent only on primary key (no transitive dependency)
 - Example:

Student:

<i>Sid</i>	<i>Sname</i>	<i>Major</i>	<i>Minor</i>	<i>IntakeYear</i>
------------	--------------	--------------	--------------	-------------------

- *Which attribute is not directly dependent on Sid?*

Student:

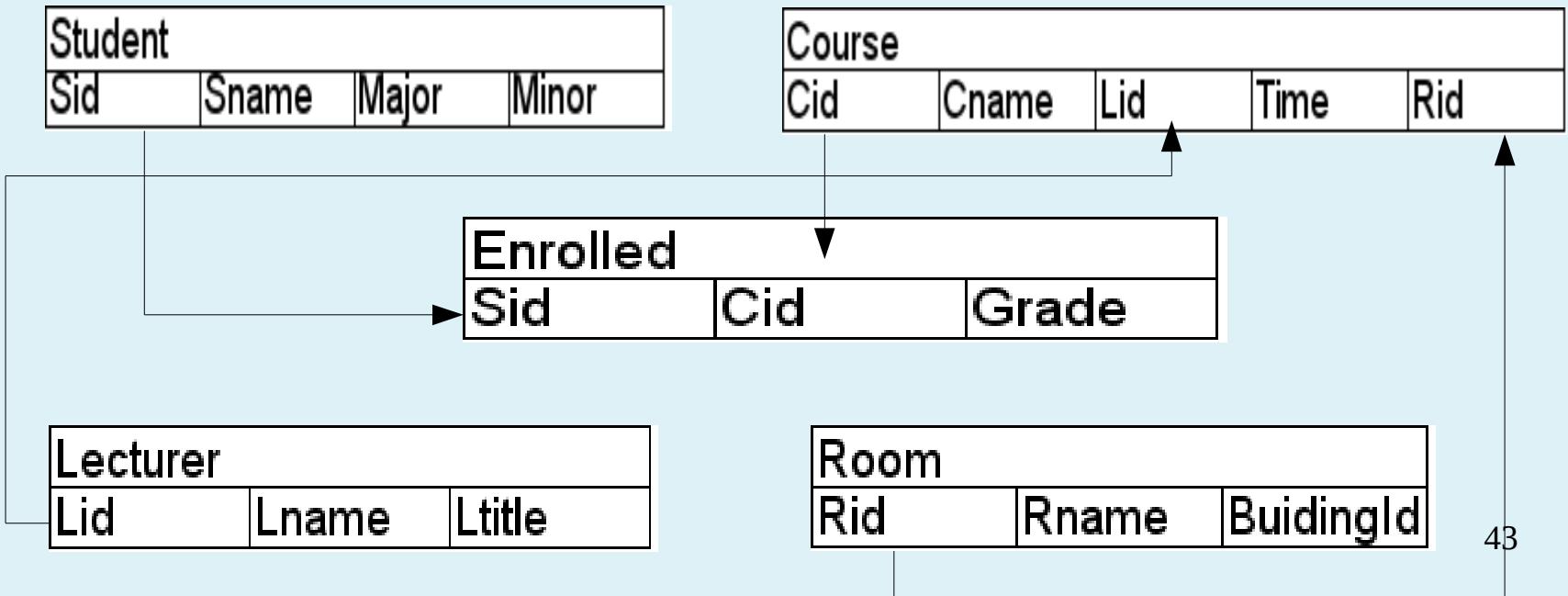
<i>Sid</i>	<i>Sname</i>	<i>Major</i>	<i>Minor</i>
------------	--------------	--------------	--------------

Database Normalization

- Old design

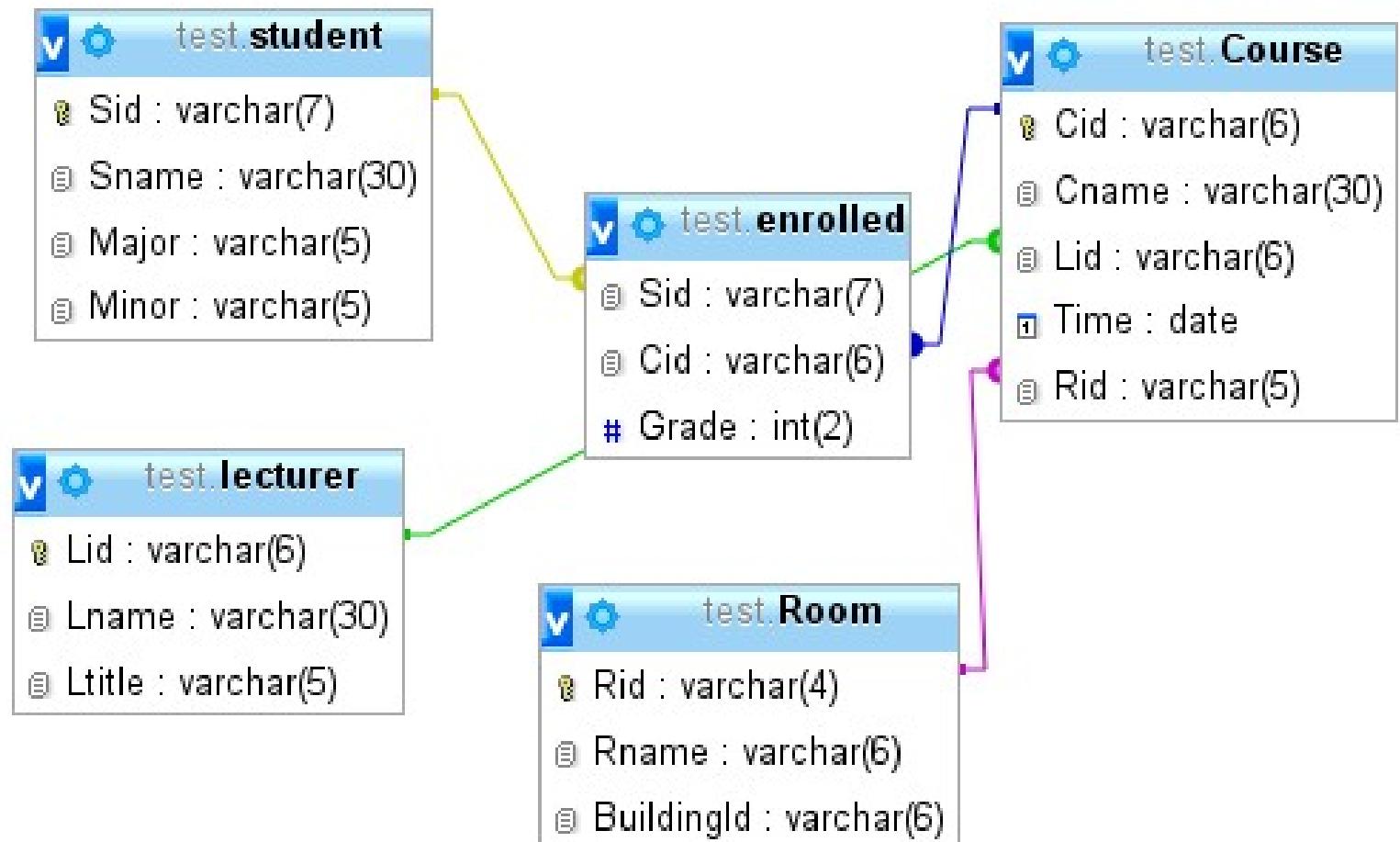
	Sid	Cname	time	room	Lid
<input type="checkbox"/>	CL0001	Machine Learning	Wed 10.15	L155	PR145
<input type="checkbox"/>	CL0002	Information Retrieval	Tue 12.15	C131	PD220
<input type="checkbox"/>	CL0003	Machine Learning	Wed 10.15	L155	PR145
<input type="checkbox"/>	CL0004	Information Extraction	Thu 10.00	C149	PR111

- New design



Database Normalization

- Storing the relation among tables in database



Database Normalization

- Exercise:
 - Which normal form does this table violate?
 - And how do you normalize it?

Person	Title	Author	Pages	Year
Yakup	Database Management System	Ramakhrisnan, Raghu	903	2010
Wenpeng	Beyond Human-Computer Interaction	Preece, Jennifer	889	2009
Amy	Support Your Local Wizard	Duane, Diane	473	1990
Dwika	The Hobbit	Tolkien, JRR	389	1995
Yadoll	Beyond Human-Computer Interaction	Preece, Jennifer	889	2009
Irina	Support Your Local Wizard	Duane, Diane	473	1990

RDBMS for Inverted Text Index

RDBMS & Full Text Searching

- Applying RDBMS for full text searching
 - What is the goal?
 - Creating an Inverted index consisting of:
 - Dictionary &
 - Posting list
 - What will be the entities?
 - Document
 - Term
 - How to start?
 - You need a specific algorithm, take for examples:
 - BSBI
 - SPIMI
 - What kind of information do you want to save in posting list?
 - Term - DocId only?
 - Term - DocId, TF, DF?

Database Design for BSBI

- A review on Blocked Sort-Based Indexing Algorithm

```
BSBINDEXCONSTRUCTION()
```

- 1 $n \leftarrow 0$
- 2 **while** (all documents have not been processed)
- 3 **do** $n \leftarrow n + 1$
- 4 $block \leftarrow \text{PARSENEXTBLOCK}()$
- 5 BSBI-INVERT($block$)
- 6 WRITEBLOCKToDISK($block, f_n$)
- 7 MERGEBLOCKS($f_1, \dots, f_n; f_{\text{merged}}$)

Database Design for BSBI

- 2 core tables:
 - Document table
 - Term tables
- How do their schemas look like?
 - Doc (did CHAR(5),
dname CHAR(6),
dcontent TEXT,
PRIMARY KEY (did), UNIQUE (dname))
 - Doc (did INT(INC),
dname CHAR(6),
dcontent BLOB,
PRIMARY KEY (did), UNIQUE (dname))
 - What are the advantages of the first scemas compared to the second or vice versa?

Database Design for BSBI

- How do their schemas look like?
 - Term (tid INT(INC),
term CHAR(25),
PRIMARY KEY (tid),
UNIQUE (term))
- The number of tables for posting list?
 - N-block tables + 1 merged posting table OR
 - 1 posting list table ?

Database Design for BSBI

Block 1

tid	did	tf
1	d2	100
2	d1	5
3	d3	57
4	d4	150

Block 2

tid	did	tf
1	d3	9
2	d4	29
5	d1	57
4	d2	82

Merged Posting

tid	did	tf
1	d2	100
1	d3	9
2	d1	5
2	d4	29
3	d3	57
4	d2	82
4	d4	150
5	d2	82

Database Design for BSBI

- The former table merging is right algorithmically, but it is a bad design in relational database. Why?
- There are several strategies for improving the design for the benefit of searching process.
- This strategy depends on the application you are developing
- Some strategies are:
 - Combining the use of file system & RDBMS for storing your data:
 - Block tables → file system
 - Merged posting list → RDBMS
 - Applying the relation & normalization concepts for merged posting list table

Database Design for BSBI

- The schema for posting list may look like as follows:
 - Posting(tid INT(), did CHAR(5), tf INT(5),
INDEX (tid, did)
FOREIGN KEY (tid, did) REFERENCES (Term,
Doc))
 - Posting(tid INT(), did STRING/TEXT(),
tf STRING/TEXT(), INDEX (tid, did)
FOREIGN KEY (tid, did) REFERENCES (Term,
Doc))
 - Posting(tid INT(), did SET(),
tf SET(), INDEX (tid, did)
FOREIGN KEY (tid, did) REFERENCES (Term,
Doc))

Database Design for SPIMI

- SPIMI differs from BSBI in:
 - The processing of dictionary → using Term instead of TermID-Term pair.
 - Memory allocation for posting list of a term.
 - Adding a posting directly to a posting list
- These differences affect little to database design.
- The former database design can be applied both to BSBI & SPIMI with one difference:
 - Term (term CHAR(25), PRIMARY KEY (term))
 - If you have only one field/column in a table, is it worth to save your data in a RDBMS?

Exercise

- Suppose you have 3 tables in your database, the dictionary (term), document (doc), and the posting list tables.
- Suppose you will compute the weight of each term using tf-idf weighting.
- How do you design your table schema for term_weight table? How do you state its relation to other tables in your database?

References

- Ramakrishnan, R. & Gehrke R. 2003. Database Management System, 2nd Ed , McGraw-Hill education.
- Delisle, M. 2006. Creating Your MySQL databases: Practical Design Tips and Techniques. Birmingham: Packt Publishing.