CS245: Databases

Introduction

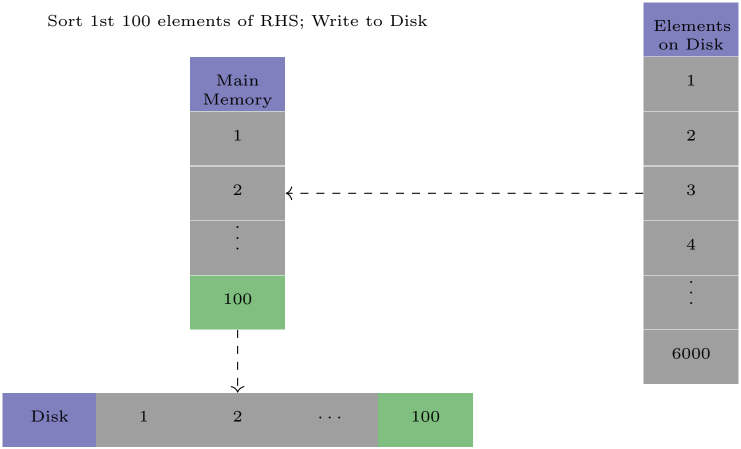
Vijaya saradhi

Department of Computer Science and Engineering
Indian Institute of Technology Guwahati

Disk based algorithms

Size and Challenges 01

Sort 1st 100 elements of RHS; Write to Disk



Main Memory

1

2

⋮

100

Elements on Disk

1

2

3

4

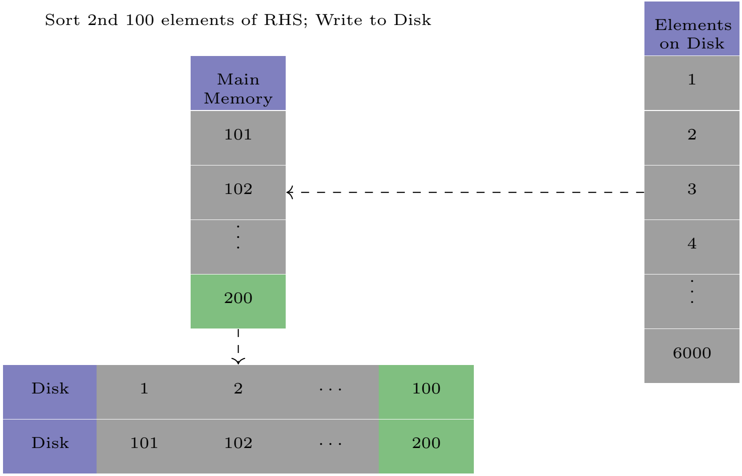
⋮

6000

Disk	1	2	· · ·	100

Size and Challenges 02

Sort 2nd 100 elements of RHS; Write to Disk



Main Memory

101

102

⋮

200

Elements on Disk

1

2

3

4

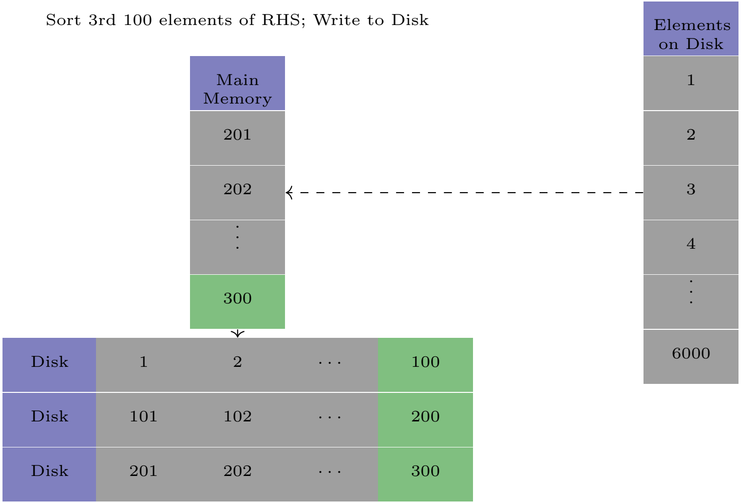
⋮

6000

Disk	1	2	· · ·	100
Disk	101	102	· · ·	200

Size and Challenges 03

Sort 3rd 100 elements of RHS; Write to Disk



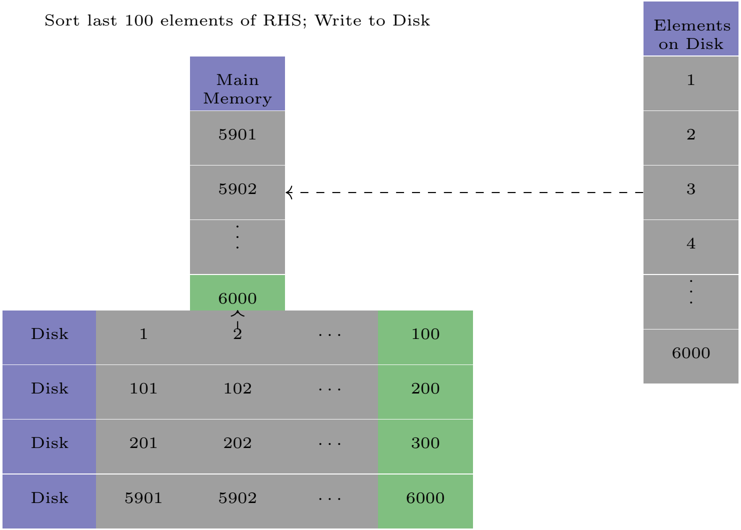
Main Memory
201
202
⋮
300

Elements on Disk
1
2
3
4
⋮
6000

Disk	1	2	· · ·	100
Disk	101	102	· · ·	200
Disk	201	202	· · ·	300

Size and Challenges 04

Sort last 100 elements of RHS; Write to Disk



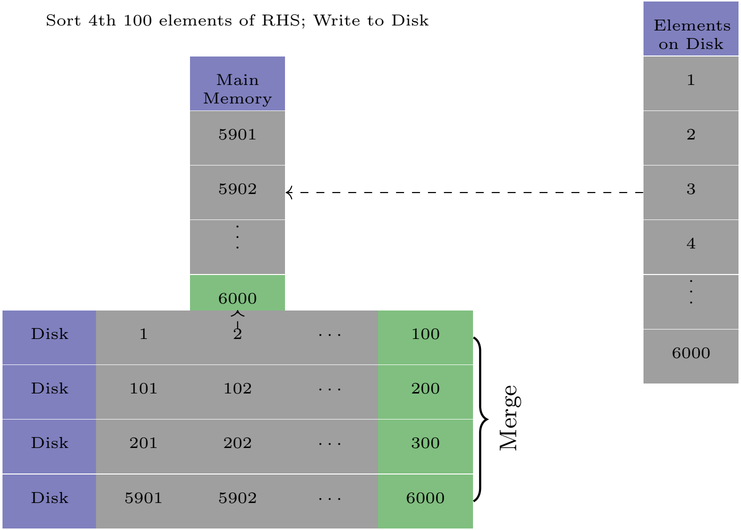
Main Memory
5901
5902
⋮
6000

Elements on Disk
1
2
3
4
⋮
6000

Disk	1	2	···	100
Disk	101	102	···	200
Disk	201	202	···	300
Disk	5901	5902	···	6000

Size and Challenges 05

Sort 4th 100 elements of RHS; Write to Disk



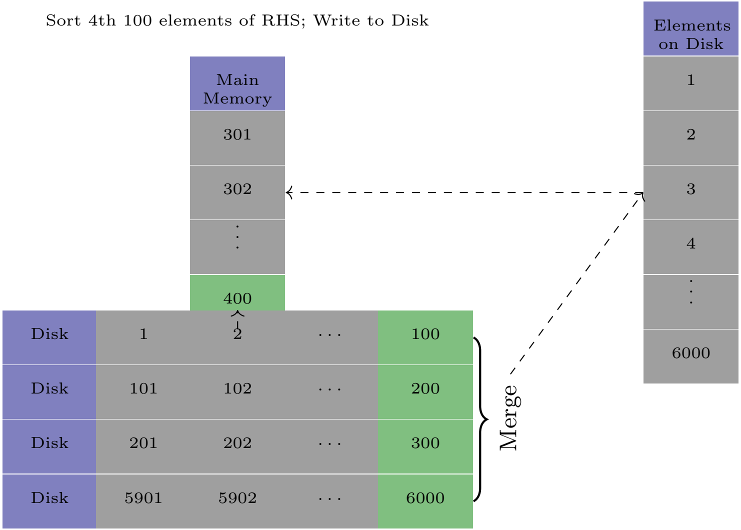
Main Memory
5901
5902
⋮
6000

Elements on Disk
1
2
3
4
⋮
6000

Disk	1	2	· · ·	100
Disk	101	102	· · ·	200
Disk	201	202	· · ·	300
Disk	5901	5902	· · ·	6000

Size and Challenges 06

Sort 4th 100 elements of RHS; Write to Disk

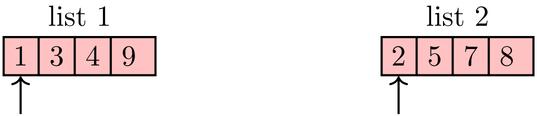


Main Memory
301
302
⋮
400

Elements on Disk
1
2
3
4
⋮
6000

Disk	1	2	· · ·	100
Disk	101	102	· · ·	200
Disk	201	202	· · ·	300
Disk	5901	5902	· · ·	6000

Merging two sorted lists 01



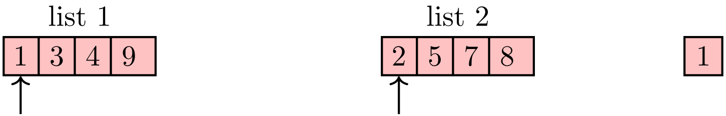
list 1

1 3 4 9

list 2

2 5 7 8

Merging two sorted lists 01



list 1

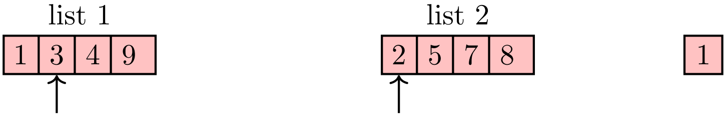
1	3	4	9

list 2

2	5	7	8

1

Merging two sorted lists 01



list 1

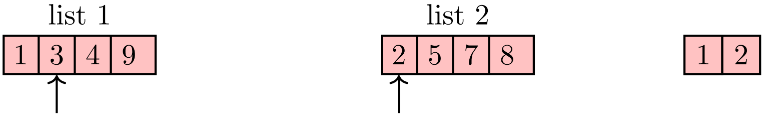
1	3	4	9

list 2

2	5	7	8

1

Merging two sorted lists 01



list 1

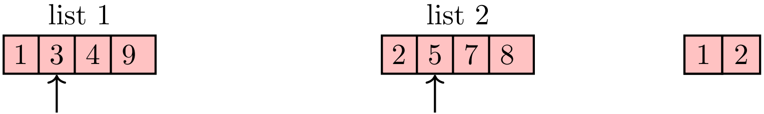
1 3 4 9

list 2

2 5 7 8

1 2

Merging two sorted lists 01



list 1

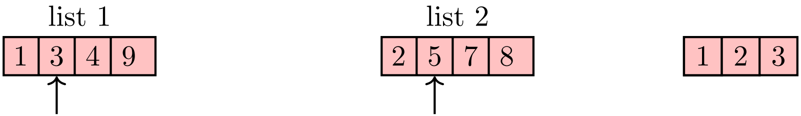
1	3	4	9

list 2

2	5	7	8

1	2

Merging two sorted lists 01



list 1

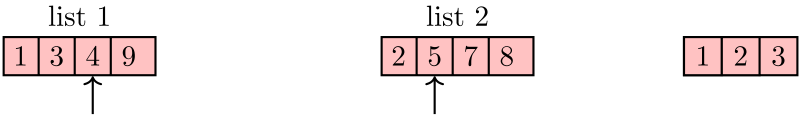
1 3 4 9

list 2

2 5 7 8

1 2 3

Merging two sorted lists 01



list 1

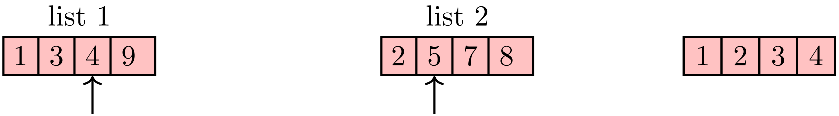
1	3	4	9

list 2

2	5	7	8

1	2	3

Merging two sorted lists 01



list 1

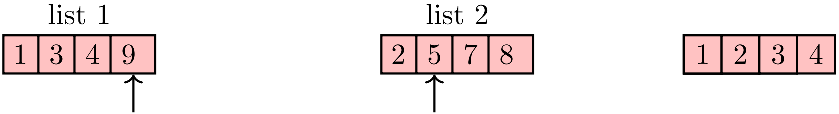
1	3	4	9

list 2

2	5	7	8

1	2	3	4

Merging two sorted lists 01



list 1

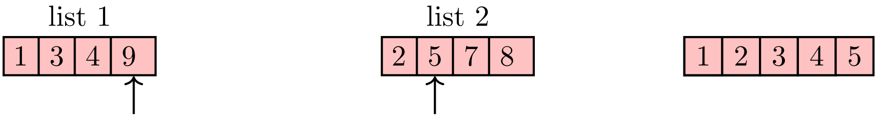
1	3	4	9

list 2

2	5	7	8

1	2	3	4

Merging two sorted lists 01



list 1

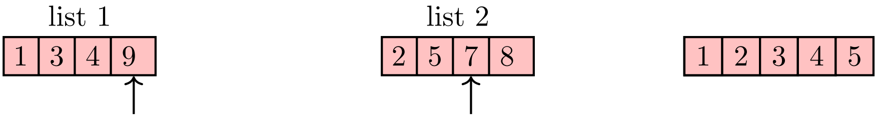
1	3	4	9

list 2

2	5	7	8

1	2	3	4	5

Merging two sorted lists 01



list 1

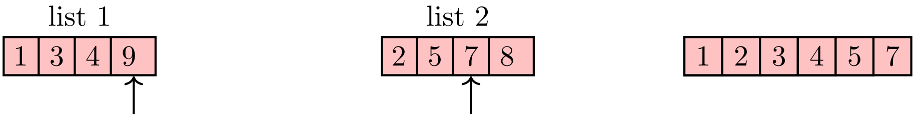
1	3	4	9

list 2

2	5	7	8

1	2	3	4	5

Merging two sorted lists 01



list 1

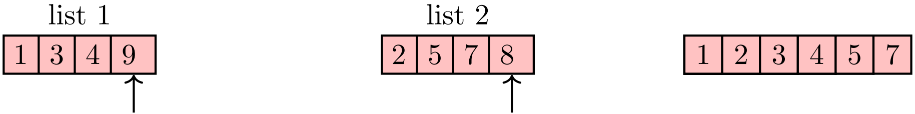
1	3	4	9

list 2

2	5	7	8

1	2	3	4	5	7

Merging two sorted lists 01



list 1

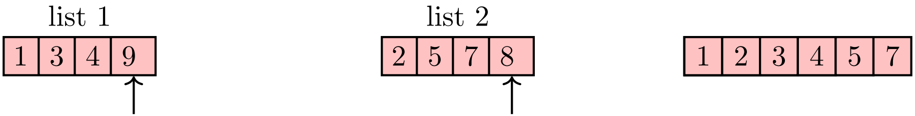
1	3	4	9

list 2

2	5	7	8

1	2	3	4	5	7

Merging two sorted lists 01



list 1

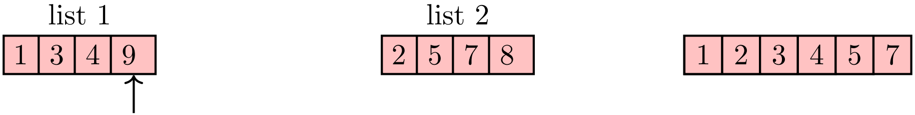
1	3	4	9

list 2

2	5	7	8

1	2	3	4	5	7

Merging two sorted lists 01



list 1

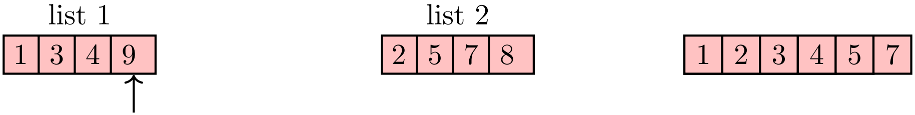
1	3	4	9

list 2

2	5	7	8

1	2	3	4	5	7

Merging two sorted lists 01



list 1

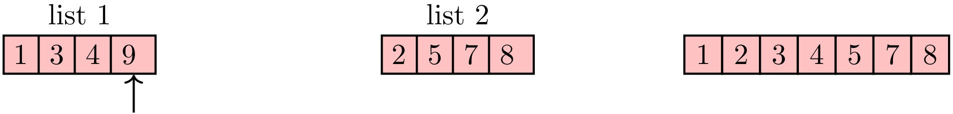
1	3	4	9

list 2

2	5	7	8

1	2	3	4	5	7

Merging two sorted lists 01



list 1

1	3	4	9

list 2

2	5	7	8

1	2	3	4	5	7	8

Merging two sorted lists 01

list 1

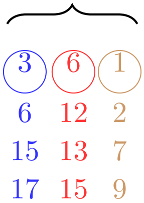
1	3	4	9

list 2

2	5	7	8

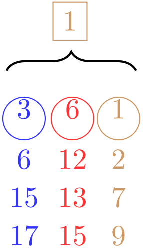
1	2	3	4	5	7	8	9

Merging multiple sorted lists 01



3	6	1
6	12	2
15	13	7
17	15	9

Merging multiple sorted lists 01



1

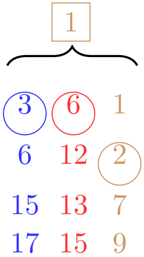
3 6 1

6 12 2

15 13 7

17 15 9

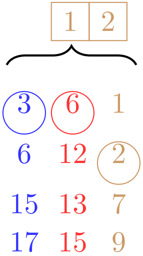
Merging multiple sorted lists 02



$\boxed{1}$

3	6	1
6	12	2
15	13	7
17	15	9

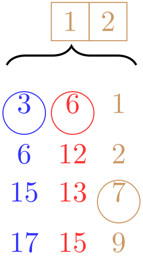
Merging multiple sorted lists 02



1	2

3	6	1
6	12	2
15	13	7
17	15	9

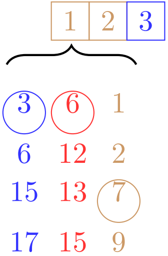
Merging multiple sorted lists 03



$\boxed{1}\boxed{2}$

3	6	1
6	12	2
15	13	7
17	15	9

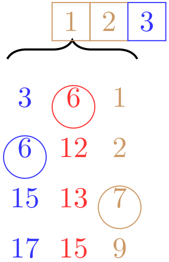
Merging multiple sorted lists 03



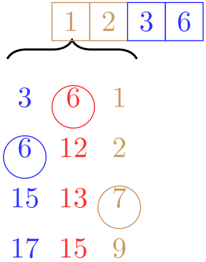
1 2 3

3	6	1
6	12	2
15	13	7
17	15	9

Merging multiple sorted lists 04



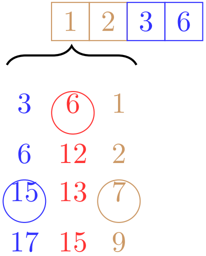
Merging multiple sorted lists 04



1	2	3	6

3	6	1
6	12	2
15	13	7
17	15	9

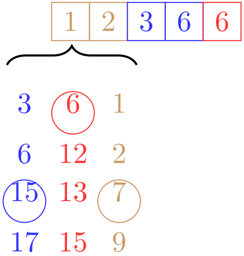
Merging multiple sorted lists 05



1	2	3	6

3	6	1
6	12	2
15	13	7
17	15	9

Merging multiple sorted lists 05



1 2 3 6 6

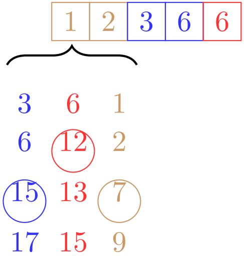
3 6 1

6 12 2

15 13 7

17 15 9

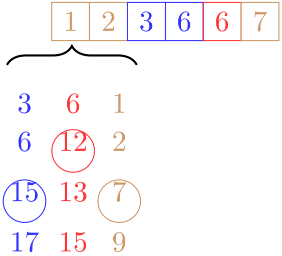
Merging multiple sorted lists 06



1	2	3	6	6

3	6	1
6	12	2
15	13	7
17	15	9

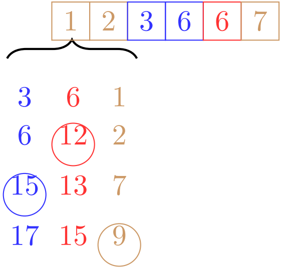
Merging multiple sorted lists 06



1	2	3	6	6	7

3	6	1
6	12	2
15	13	7
17	15	9

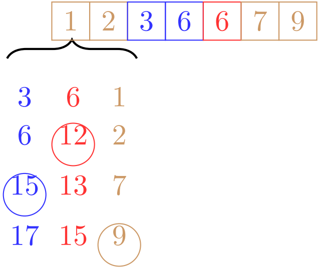
Merging multiple sorted lists 07



1	2	3	6	6	7

3	6	1
6	12	2
15	13	7
17	15	9

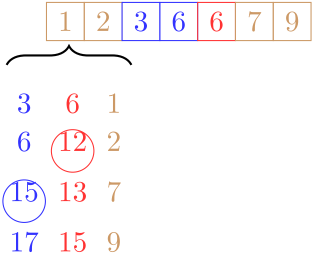
Merging multiple sorted lists 07



1	2	3	6	6	7	9

3	6	1
6	12	2
15	13	7
17	15	9

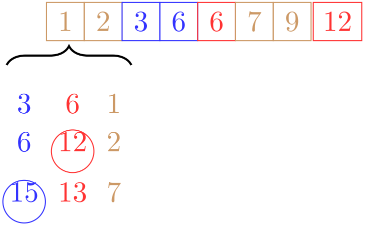
Merging multiple sorted lists 08



1	2	3	6	6	7	9

3	6	1
6	12	2
15	13	7
17	15	9

Merging multiple sorted lists 08



1 2 3 6 6 7 9 12

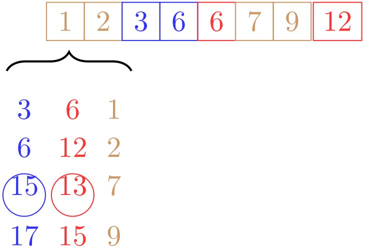
3 6 1

6 12 2

15 13 7

17 15 9

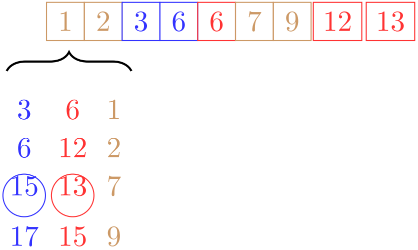
Merging multiple sorted lists 09



1	2	3	6	6	7	9	12

3	6	1
6	12	2
15	13	7
17	15	9

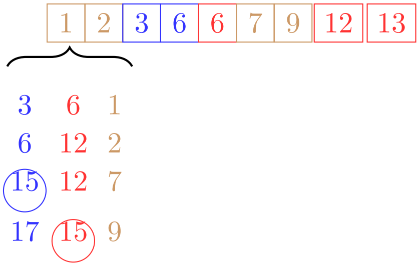
Merging multiple sorted lists 09



1	2	3	6	6	7	9	12	13

3	6	1
6	12	2
15	13	7
17	15	9

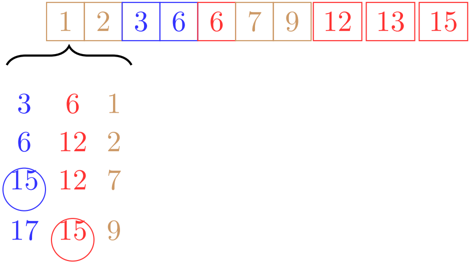
Merging multiple sorted lists 10



1	2	3	6	6	7	9	12	13

3	6	1
6	12	2
15	12	7
17	15	9

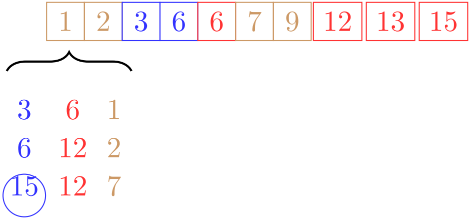
Merging multiple sorted lists 10



1	2	3	6	6	7	9	12	13	15

3	6	1
6	12	2
15	12	7
17	15	9

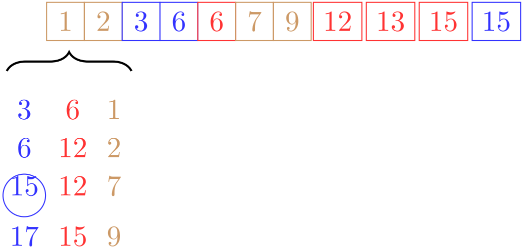
Merging multiple sorted lists 11



1	2	3	6	6	7	9	12	13	15

3	6	1
6	12	2
15	12	7
17	15	9

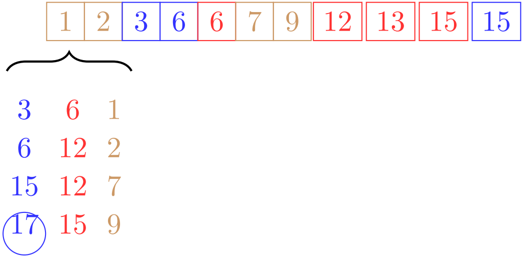
Merging multiple sorted lists 11



1	2	3	6	6	7	9	12	13	15	15

3	6	1
6	12	2
15	12	7
17	15	9

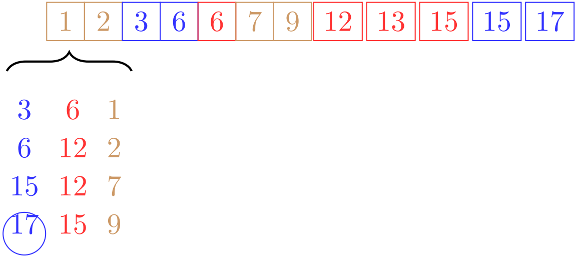
Merging multiple sorted lists 12



1	2	3	6	6	7	9	12	13	15	15

3	6	1
6	12	2
15	12	7
17	15	9

Merging multiple sorted lists 12

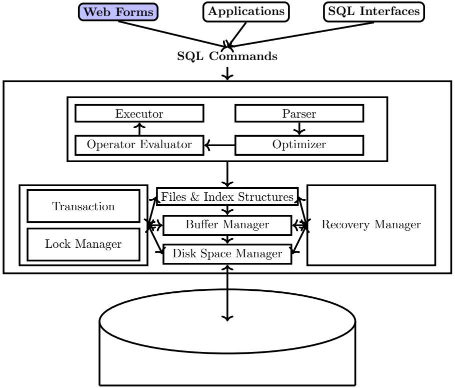


1	2	3	6	6	7	9	12	13	15	15	17

3	6	1
6	12	2
15	12	7
17	15	9

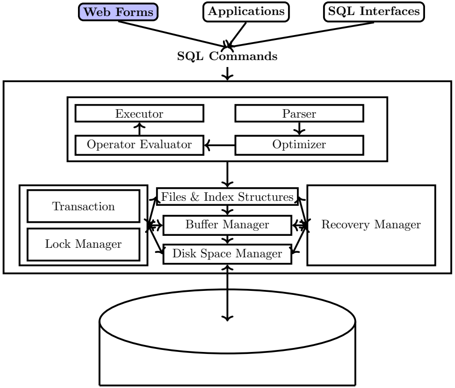
Database management system architecture

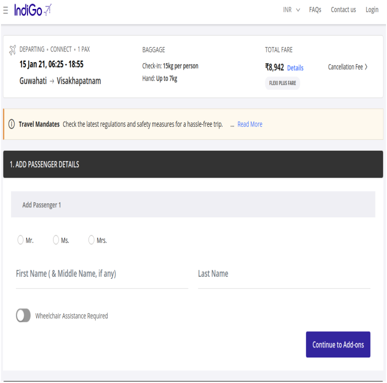
DBMS Architecture



For Students RollNo is the LoginId
For Faculty members email Id without
@iitg.ac.in is LoginId

DBMS Architecture





DEPARTING • CONNECT • 1 PAX

15 Jan 21, 06:25 - 18:55

Guwahati → Visakhapatnam

BAGGAGE

Check-in: 15kg per person

Hand: Up to 7kg

TOTAL FARE

₹8,942

FLEXI PLUS FARE

Travel Mandates Check the latest regulations and safety measures for a hassle-free trip. ...

1. ADD PASSENGER DETAILS

Add Passenger 1

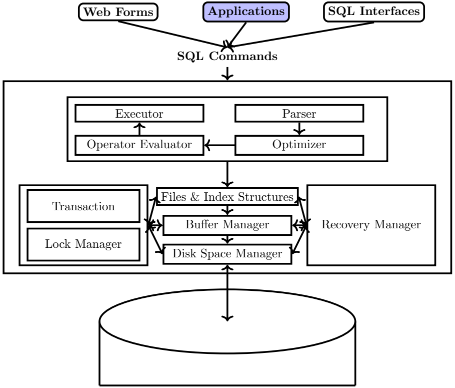
○ Mr. ○ Ms. ○ Mrs.

First Name (& Middle Name, if any)

Last Name

Wheelchair Assistance Required

DBMS Architecture



Web Forms

Applications

SQL Interfaces

SQL Commands

Executor

Parser

Operator Evaluator

Optimizer

Files & Index Structures

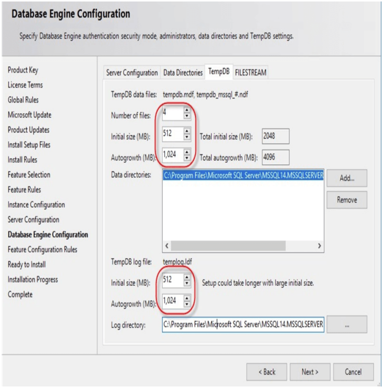
Transaction

Lock Manager

Buffer Manager

Recovery Manager

Disk Space Manager



Database Engine Configuration

Specify Database Engine authentication security mode, administrators, data directories and TempDB settings.

TempDB data files: tempdb.mdf, tempdb_mssql_#.ndf

Number of files: 4

Initial size (MB): 512 Total initial size (MB): 2048

Autogrowth (MB): 1,024 Total autogrowth (MB): 4096

Data directories: C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER

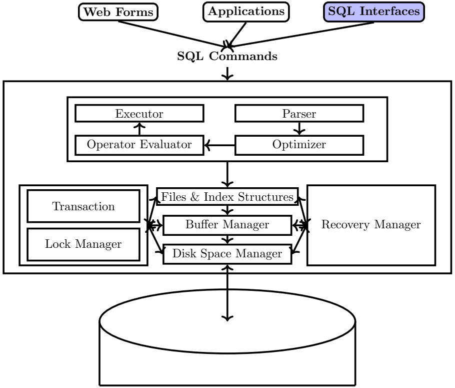
TempDB log file: templog.ldf

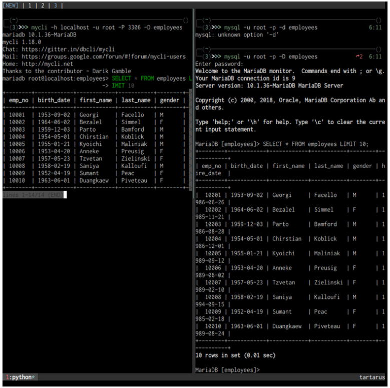
Initial size (MB): 512 Setup could take longer with large initial size.

Autogrowth (MB): 1,024

Log directory: C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER

DBMS Architecture

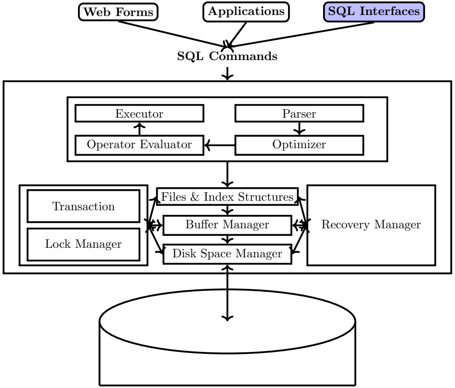


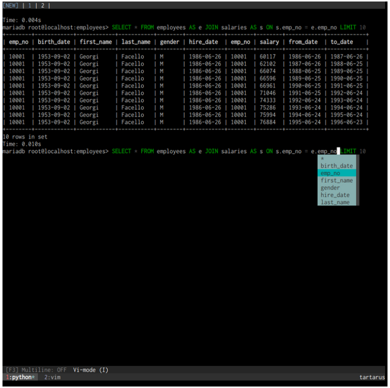


```
(3)>>> mycli -h localhost -u root -P 3306 -D employees
mariadb 10.1.36-MariaDB
mycli 1.18.0
Chat: https://gitter.im/dbcli/mycli
Mail: https://groups.google.com/forum/#!forum/mycli-users
Home: http://mycli.net
Thanks to the contributor - Darik Gamble
mariadb root@localhost:employees> SELECT * FROM employees L
                               -> IMIT 10
+--------+------------+------------+-----------+--------+
| emp_no | birth_date | first_name | last_name | gender |
+--------+------------+------------+-----------+--------+
| 10001  | 1953-09-02 | Georgi     | Facello   | M      |
| 10002  | 1964-06-02 | Bezalel    | Simmel    | F      |
| 10003  | 1959-12-03 | Parto      | Bamford   | M      |
| 10004  | 1954-05-01 | Chirstian  | Koblick   | M      |
| 10005  | 1955-01-21 | Kyoichi    | Maliniak  | M      |
| 10006  | 1953-04-20 | Anneke     | Preusig   | F      |
| 10007  | 1957-05-23 | Tzvetan    | Zielinski | F      |
| 10008  | 1958-02-19 | Saniya     | Kalloufi  | M      |
| 10009  | 1952-04-19 | Sumant     | Peac      | F      |
| 10010  | 1963-06-01 | Duangkaew  | Piveteau  | F      |
+--------+------------+------------+-----------+--------+
lines 1-14/14 (END)
```

```
(3)>>> mysql -u root -p -d employees                    6:11
mysql: unknown option '-d'

(3)>>> mysql -u root -p -D employees                 2  6:11
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 9
Server version: 10.1.36-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab an
d others.

Type 'help;' or '\h' for help. Type '\c' to clear the curre
nt input statement.

MariaDB [employees]> SELECT * FROM employees LIMIT 10;
+--------+------------+------------+-----------+--------+--
----------+
| emp_no | birth_date | first_name | last_name | gender | h
ire_date  |
+--------+------------+------------+-----------+--------+--
----------+
|  10001 | 1953-09-02 | Georgi     | Facello   | M      | 1
986-06-26 |
|  10002 | 1964-06-02 | Bezalel    | Simmel    | F      | 1
985-11-21 |
|  10003 | 1959-12-03 | Parto      | Bamford   | M      | 1
986-08-28 |
|  10004 | 1954-05-01 | Chirstian  | Koblick   | M      | 1
986-12-01 |
|  10005 | 1955-01-21 | Kyoichi    | Maliniak  | M      | 1
989-09-12 |
|  10006 | 1953-04-20 | Anneke     | Preusig   | F      | 1
989-06-02 |
|  10007 | 1957-05-23 | Tzvetan    | Zielinski | F      | 1
989-02-10 |
|  10008 | 1958-02-19 | Saniya     | Kalloufi  | M      | 1
994-09-15 |
|  10009 | 1952-04-19 | Sumant     | Peac      | F      | 1
985-02-18 |
|  10010 | 1963-06-01 | Duangkaew  | Piveteau  | F      | 1
989-08-24 |
+--------+------------+------------+-----------+--------+--
----------+
10 rows in set (0.01 sec)

MariaDB [employees]>
```

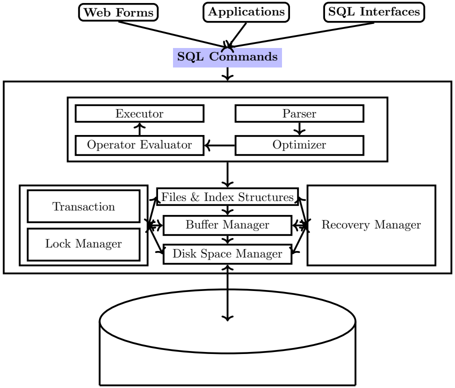
DBMS Architecture





```
[NEW] | 1 | 2 |

Time: 0.004s
mariadb root@localhost:employees> SELECT * FROM employees AS e JOIN salaries AS s ON s.emp_no = e.emp_no LIMIT 10
+--------+------------+------------+-----------+--------+------------+--------+--------+------------+------------+
| emp_no | birth_date | first_name | last_name | gender | hire_date  | emp_no | salary | from_date  | to_date    |
+--------+------------+------------+-----------+--------+------------+--------+--------+------------+------------+
| 10001  | 1953-09-02 | Georgi     | Facello   | M      | 1986-06-26 | 10001  | 60117  | 1986-06-26 | 1987-06-26 |
| 10001  | 1953-09-02 | Georgi     | Facello   | M      | 1986-06-26 | 10001  | 62102  | 1987-06-26 | 1988-06-25 |
| 10001  | 1953-09-02 | Georgi     | Facello   | M      | 1986-06-26 | 10001  | 66074  | 1988-06-25 | 1989-06-25 |
| 10001  | 1953-09-02 | Georgi     | Facello   | M      | 1986-06-26 | 10001  | 66596  | 1989-06-25 | 1990-06-25 |
| 10001  | 1953-09-02 | Georgi     | Facello   | M      | 1986-06-26 | 10001  | 66961  | 1990-06-25 | 1991-06-25 |
| 10001  | 1953-09-02 | Georgi     | Facello   | M      | 1986-06-26 | 10001  | 71046  | 1991-06-25 | 1992-06-24 |
| 10001  | 1953-09-02 | Georgi     | Facello   | M      | 1986-06-26 | 10001  | 74333  | 1992-06-24 | 1993-06-24 |
| 10001  | 1953-09-02 | Georgi     | Facello   | M      | 1986-06-26 | 10001  | 75286  | 1993-06-24 | 1994-06-24 |
| 10001  | 1953-09-02 | Georgi     | Facello   | M      | 1986-06-26 | 10001  | 75994  | 1994-06-24 | 1995-06-24 |
| 10001  | 1953-09-02 | Georgi     | Facello   | M      | 1986-06-26 | 10001  | 76884  | 1995-06-24 | 1996-06-23 |
+--------+------------+------------+-----------+--------+------------+--------+--------+------------+------------+

10 rows in set
Time: 0.010s
mariadb root@localhost:employees> SELECT * FROM employees AS e JOIN salaries AS s ON s.emp_no = e.emp_no LIMIT 10
                                                                                           *
                                                                                           birth_date
                                                                                           emp_no
                                                                                           first_name
                                                                                           gender
                                                                                           hire_date
                                                                                           last_name

[F3] Multiline: OFF  Vi-mode (I)
1:python*  2:vim                                                                                          tartarus
```

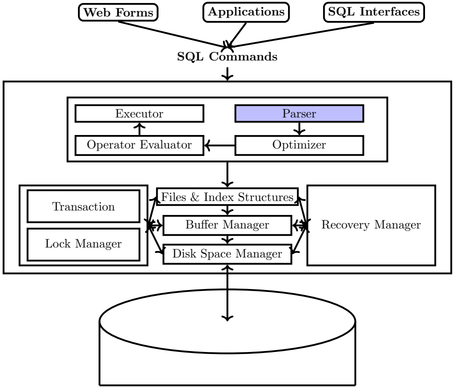
DBMS Architecture

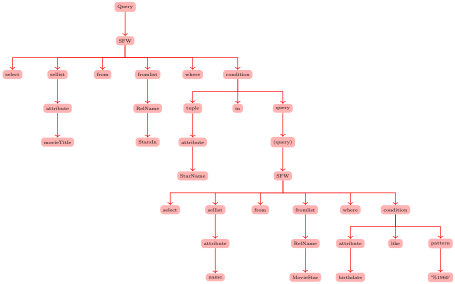


SQL Statement

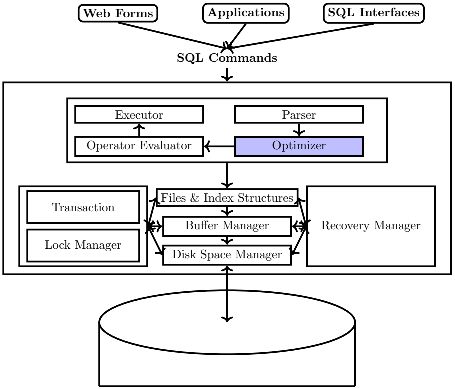
```
 SELECT movieTitle
FROM StarsIn, MovieStar
WHERE starName = name AND
birthdate LIKE '%1960';
```

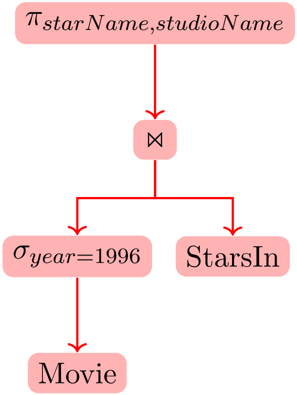
DBMS Architecture



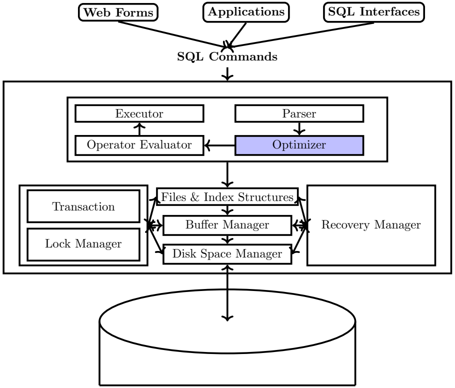


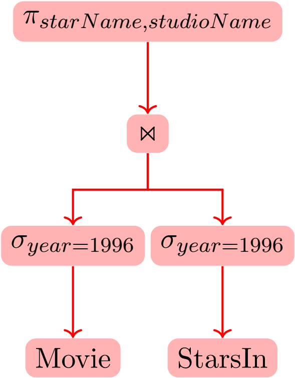
DBMS Architecture



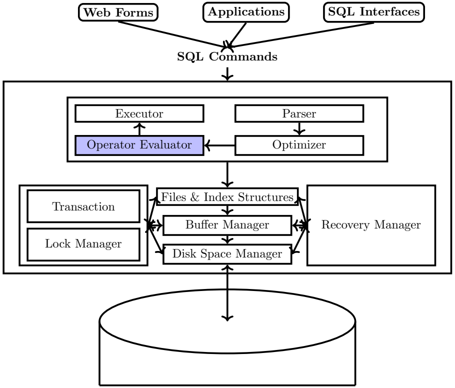


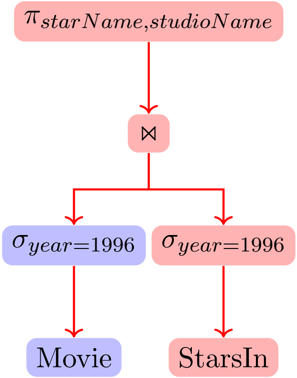
DBMS Architecture



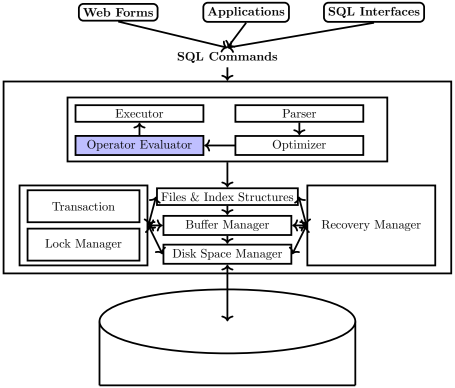


DBMS Architecture





DBMS Architecture



Web Forms

Applications

SQL Interfaces

SQL Commands

Executor

Parser

Operator Evaluator

Optimizer

Transaction

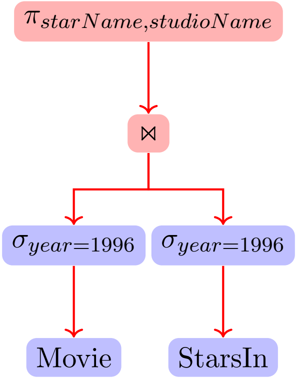
Lock Manager

Files & Index Structures

Buffer Manager

Disk Space Manager

Recovery Manager



$\pi_{starName,studioName}$

$\bowtie$

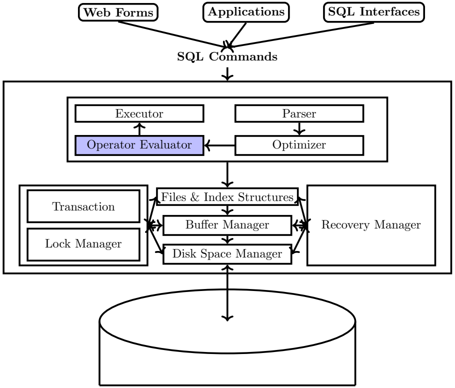
$\sigma_{year=1996}$

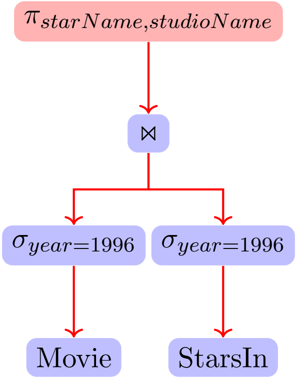
$\sigma_{year=1996}$

Movie

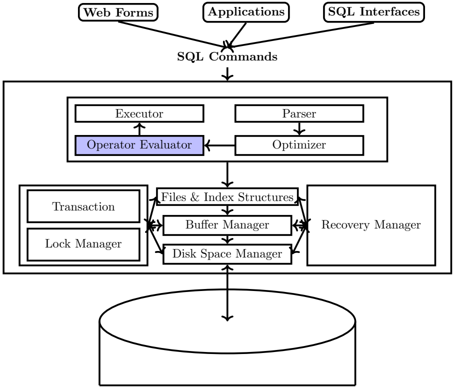
StarsIn

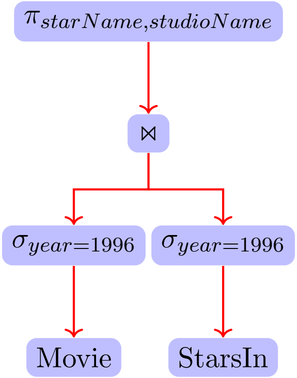
DBMS Architecture



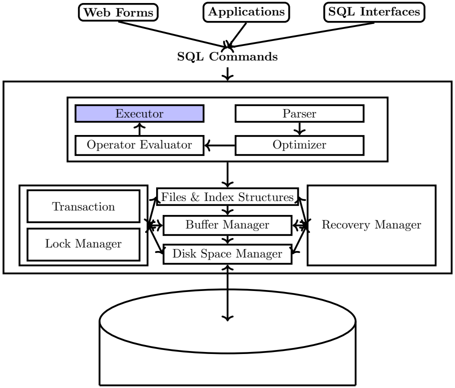


DBMS Architecture

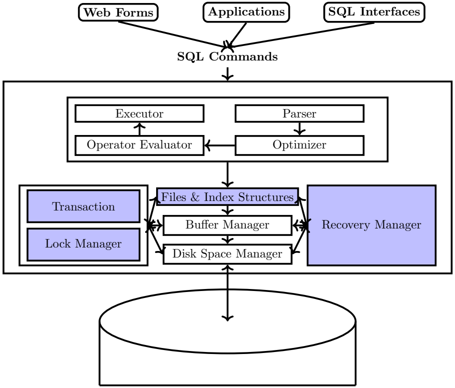




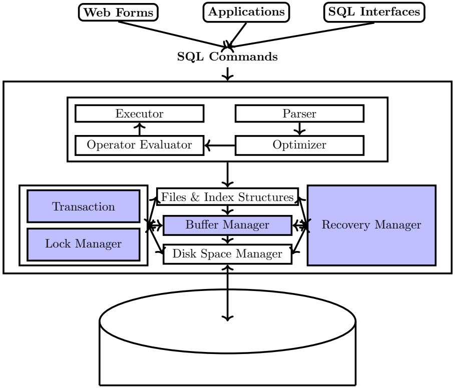
DBMS Architecture



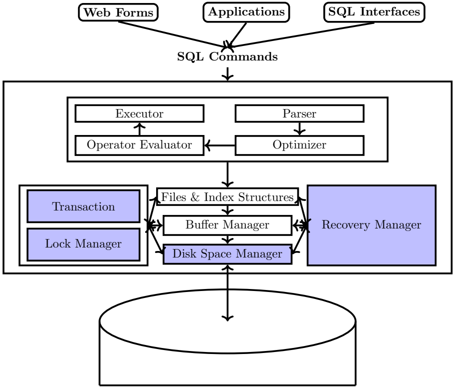
DBMS Architecture



DBMS Architecture

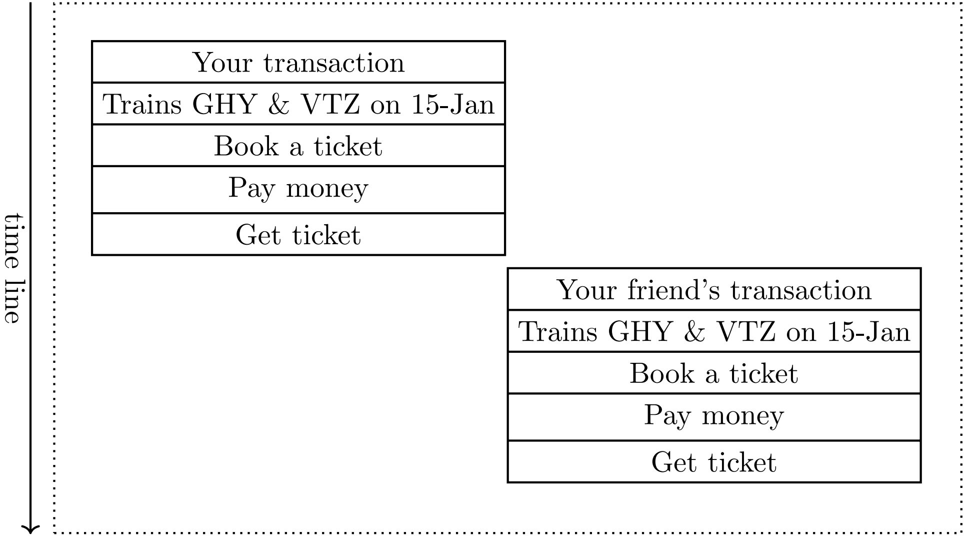


DBMS Architecture



Transactions

Transaction 01



time line

Your transaction

Trains GHY & VTZ on 15-Jan

Book a ticket

Pay money

Get ticket

Your friend's transaction

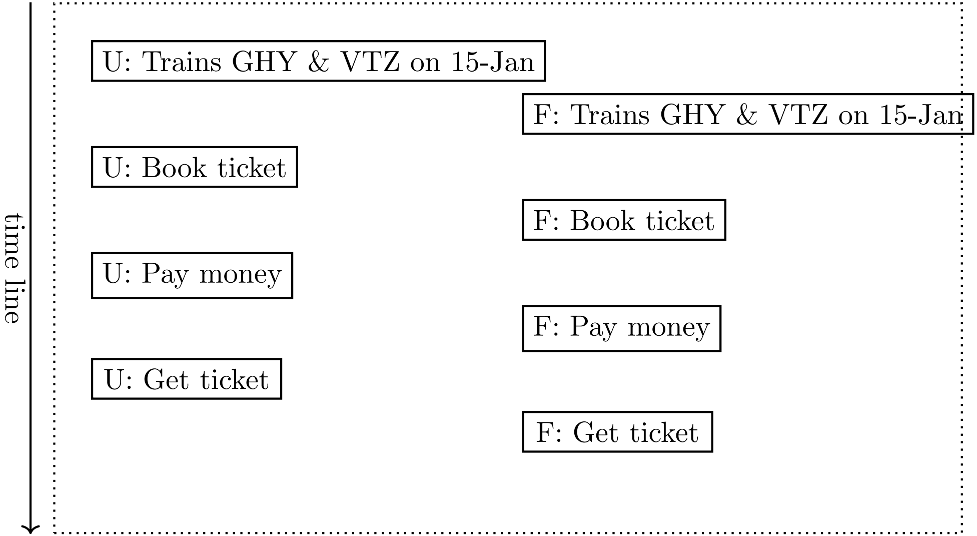
Trains GHY & VTZ on 15-Jan

Book a ticket

Pay money

Get ticket

Transaction 02



U: Trains GHY & VTZ on 15-Jan

F: Trains GHY & VTZ on 15-Jan

U: Book ticket

F: Book ticket

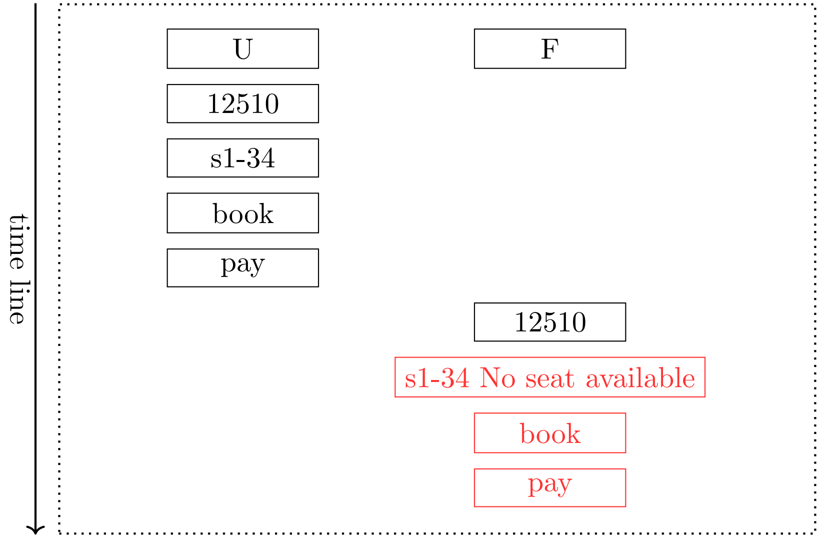
U: Pay money

F: Pay money

U: Get ticket

F: Get ticket

Transaction 03



time line

U

12510

s1-34

book

pay

F

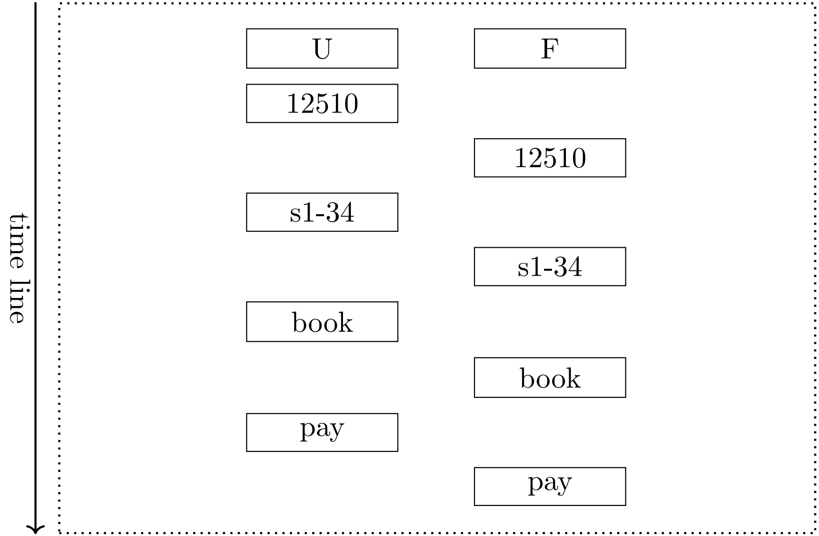
12510

s1-34 No seat available

book

pay

Transaction 04

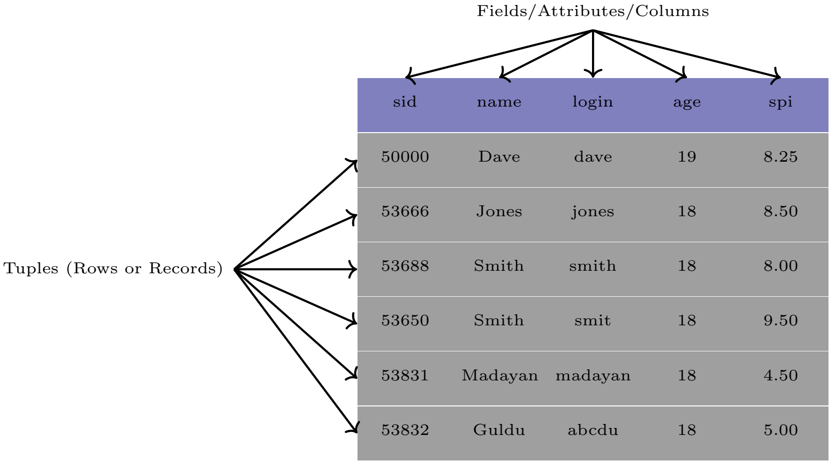


time line

U	F
12510	
	12510
s1-34	
	s1-34
book	
	book
pay	
	pay

Tables

Table Notations



Fields/Attributes/Columns

Tuples (Rows or Records)

sid	name	login	age	spi
50000	Dave	dave	19	8.25
53666	Jones	jones	18	8.50
53688	Smith	smith	18	8.00
53650	Smith	smit	18	9.50
53831	Madayan	madayan	18	4.50
53832	Guldu	abcdu	18	5.00

Need for constraints on table

student				
sid	name	login	age	spi
190101000	Atul Kumar	atul	18	8.0
190101000	Atul Gupta	atul	18	8.2
190101000	Atul M	atul	18	8.2
190101000	Atul Gupta	atul	19	7.2

- Same roll number is assigned to several students
- Same login is assigned to several students
- It is not possible to distinguish between two Atul Gupta's (row 2 & 4)
- In case you have to update the spi of Atul Gupta which row will you update? 2 or 4?

Table Notations

Constraints on Tables

Not discussing all the constraints at present

- Primary key
- Uniqueness
- Not NULL
- DEFAULT
- (requires) Two or more tables - Foreign key

Primary key - 01

single column

- One column designated as primary key
- When primary key column has identical values then corresponding rows must have identical values OR
- Primay key values must be all distinct

Primary key - 01

single column

- One column designated as primary key
- When primary key column has identical values then corresponding rows must have identical values OR
- Primay key values must be all distinct

single column - example 01

sid is single column primary key

Violation of primary key constraint

student

sid	name	login	age	spi
190101000	Atul Kumar	atul	18	8.0
190101001	Atul Gupta	atul	18	8.2
190101001	Atul M	atul	18	8.2
190101002	Atul Gupta	atul	19	7.2

Primary key - 01

single column

- One column designated as primary key
- When primary key column has identical values then corresponding rows must have identical values OR
- Primay key values must be all distinct

single column - example 02

sid is single column primary key
No violation of primary key constraint
However, database engines will not allow two identical values in primary key column

student

sid	name	login	age	spi
190101000	Atul Kumar	atul	18	8.0
190101000	Atul Kumar	atul	18	8.0
190101001	Atul M	atul	18	8.2
190101002	Atul Gupta	atul	19	7.2

Primary key - 02

Two columns

- Two column combindly described as primary key
- When primary key **columns** has identical values then corresponding rows must have identical values OR
- Primay key values must be all distinct

Primary key - 02

Two columns

- Two column combindly described as primary key
- When primary key **columns** has identical values then corresponding rows must have identical values OR
- Primay key values must be all distinct

two columns - example 01

{**sid, cid**} together are primary key

Violation of primary key constraint

register

sid	grade	**cid**
190101000	AB	CS101
190101000	BB	CS101
190109001	AA	CS101
190109001	BB	CS102

Primary key - 02

Two columns

- Two column combindly described as primary key
- When primary key **columns** has identical values then corresponding rows must have identical values OR
- Primary key values must be all distinct

two column - example 02

{**sid, cid**} together are primary key

No violation of primary key constraint

However, database engines will not allow two identical values in primary key column

register

sid	grade	**cid**
190101000	AB	CS101
190101000	AB	CS101
190109001	AA	CS101
190109001	BB	CS102

Primary key - 03

All columns

- All the columns combindly described as primary key
- When primary key **columns** has identical values then corresponding rows must have identical values OR
- Primay key values must be all distinct

Primary key - 03

All columns

- All the columns combindly described as primary key
- When primary key **columns** has identical values then corresponding rows must have identical values OR
- Primay key values must be all distinct

All columns - example 01

{**sid, year, cid**} together are primary key
Is this primary key constraint violation?

register

sid	**year**	**cid**
190101000	2020	CS101
190101000	2020	CS101
190109001	2021	CS101
190109001	2022	CS102

Primary key - 03

All columns

- All the columns combindly described as primary key
- When primary key **columns** has identical values then corresponding rows must have identical values OR
- Primay key values must be all distinct

All columns - example 02

{**sid, year, cid**} together are primary key
Is this primary key constraint violation?

register

sid	**year**	**cid**
190101000	2020	CS101
190101000	2021	CS101
190109001	2021	CS101
190109001	2022	CS102

Primary key - 04

Why Primary?

In the student table example

- **sid** is a key.
- login also can be a key.
- No two students can have identical login values.
- We choose one of them to be the primary key
- All queries use to **sid** for convenience
- It is possible that queries may use login key instead of primary key to retrieve data

Primary key - 04a

student				
sid	name	login	age	spi
190101001	Atul Kumar	atul	18	8.0
190101002	Atul Kumar	ak	18	8.0
190101003	Atul M	atulm	18	8.2
190101004	Atul Gupta	atulg	19	7.2

Primary key - 04a

student				
sid	name	login	age	spi
190101001	Atul Kumar	atul	18	8.0
190101002	Atul Kumar	ak	18	8.0
190101003	Atul M	atulm	18	8.2
190101004	Atul Gupta	atulg	19	7.2

example - 01

- What is the spi of student with **sid** 190101001?

Primary key - 04a

student

sid	name	login	age	spi
190101001	Atul Kumar	atul	18	8.0
190101002	Atul Kumar	ak	18	8.0
190101003	Atul M	atulm	18	8.2
190101004	Atul Gupta	atulg	19	7.2

example - 01

- What is the spi of student with **sid** 190101001?
- What is the spi of student with login atul?

Primary key - 04a

student

sid	name	login	age	spi
190101001	Atul Kumar	atul	18	8.0
190101002	Atul Kumar	ak	18	8.0
190101003	Atul M	atulm	18	8.2
190101004	Atul Gupta	atulg	19	7.2

example - 01

- What is the spi of student with **sid** 190101001?
- What is the spi of student with login atul?
- Can you query: What is the spi of student with name “Atul Kumar”?

Primary key - 04a

student

sid	name	login	age	spi
190101001	Atul Kumar	atul	18	8.0
190101002	Atul Kumar	ak	18	8.0
190101003	Atul M	atulm	18	8.2
190101004	Atul Gupta	atulg	19	7.2

example - 01

- What is the spi of student with **sid** 190101001?
- What is the spi of student with login atul?
- Can you query: What is the spi of student with name "Atul Kumar"?
- Can you query: What is the spi of student with age 18?

Primary key - 04a

student

sid	name	login	age	spi
190101001	Atul Kumar	atul	18	8.0
190101002	Atul Kumar	ak	18	8.0
190101003	Atul M	atulm	18	8.2
190101004	Atul Gupta	atulg	19	7.2

example - 01

- What is the spi of student with **sid** 190101001?
- What is the spi of student with login atul?
- Can you query: What is the spi of student with name “Atul Kumar”?
- Can you query: What is the spi of student with age 18?
- Last two queries are not erronous. They result in retrieving multiple rows.

Identifier (keys) - 01

Description

- Exists purely to identify rows of a table (relation/entity)
- Do not imply any property of instances
- Example: Order number, product code, batch number, etc.

Identifier (keys)- 02

Details

IDs may be of three types

- System generated
- Administrator generated
- Externally defined identifiers

System Generated

Examples

- Order numbers (no human intervention)
- Account numbers, RD number, FD number, mobile number, etc..
- Generated in sequence without any specific requirement of the sequence generation
- Can be numeric and non-numeric

Administrator Generated

Examples

- Only suitable for relatively low-volume entity classes
- Department codes, product codes, class room numbers, course codes etc
- Can be numeric or non-numeric
- Administrator have mechanism to create new identifiers

Externally Defined

Examples

- Defined by external party
- Often by national or international standards authority
- Country codes (telephone numbers)
- Currency codes
- State codes
- Pin codes
- Codes externally defined but administrator generated for postal department

Identifiers

Role

- Used in many instances of operations
- Used as constraints
- Uniquely identifying rows of a table

Primary key - 05

Primary Key

sid	name	login	age	spi
50000	Dave	dave	19	8.25
53666	Jones	jones	18	8.50
53688	Smith	smith	18	8.00
53650	Smith	smit	18	9.50
53831	Madayan	madayan	18	4.50
53832	Guldu	abcdu	18	5.00

Primary key - 06

Second row is legal when there is no constraint on `login` column

Primary Key

sid	name	login	age	spi
50000	Dave	dave	19	8.25
50001	Dave	dave	19	8.25
53666	Jones	jones	18	8.50
53688	Smith	smith	18	8.00
53650	Smith	smit	18	9.50
53831	Madayan	madayan	18	4.50
53832	Guldu	abcdu	18	5.00

Primary key violation - 07

Cannot have two rows having identical `sid` values

Primary Key

↓

sid	name	login	age	spi
50000	Dave	dave	19	8.25
53666	Jones	jones	18	8.50
53688	Smith	smith	18	8.00
53650	Smith	smit	18	9.50
53831	Madayan	madayan	18	4.50
53832	Guldu	abcdu	18	5.00
53666	James	james	18	8.50

Primary key - 08

More than one column can participate in Primary Key

Primary Key　　　Primary Key

cid	grade	sid
CS101	CC	53831
CS203	BB	53832
CS112	AB	53650
HS105	BB	53666

Primary key - 09

Relation between course table and student table

Primary Key Primary Key

cid	grade	sid
CS101	CC	53831
CS203	BB	53832
CS112	AB	53650
HS105	BB	53666
HS106	AB	53666

One student registering for two courses

Primary key - 10

Relation between course table and student table

Primary Key Primary Key

cid	grade	sid
CS101	CC	53831
CS203	BB	53832
CS112	AB	53650
HS105	BB	53666
HS105	BB	53832

Two students registering for one course

Primary key - 11

Relationship between three tables

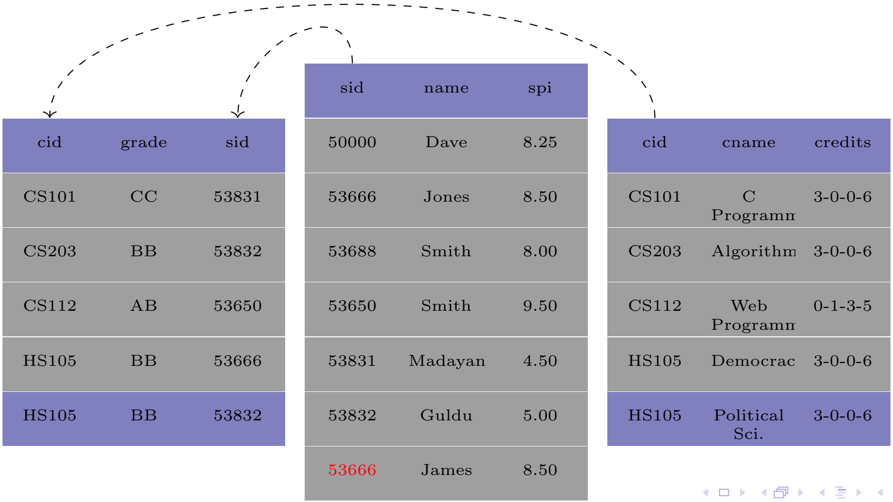
Register		
cid	grade	sid
CS101	CC	53831
CS203	BB	53832
CS112	AB	53650
HS105	BB	53666
HS105	BB	53832

Student		
sid	name	spi
50000	Dave	8.25
53666	Jones	8.50
53688	Smith	8.00
53650	Smith	9.50
53831	Madayan	4.50
53832	Guldu	5.00
53666	James	8.50

Course		
cid	cname	credits
CS101	C Programm	3-0-0-6
CS203	Algorithm	3-0-0-6
CS112	Web Programm	0-1-3-5
HS105	Economic	3-0-0-6
HS105	Political Sci.	3-0-0-6

Primary key - 12

Relationship between three tables



cid	grade	sid
CS101	CC	53831
CS203	BB	53832
CS112	AB	53650
HS105	BB	53666
HS105	BB	53832

sid	name	spi
50000	Dave	8.25
53666	Jones	8.50
53688	Smith	8.00
53650	Smith	9.50
53831	Madayan	4.50
53832	Guldu	5.00
53666	James	8.50

cid	cname	credits
CS101	C Programn	3-0-0-6
CS203	Algorithm	3-0-0-6
CS112	Web Programn	0-1-3-5
HS105	Democrac	3-0-0-6
HS105	Political Sci.	3-0-0-6

Primary key - 13 violation

None of the columns of the primary key should have ⊥ values

Primary Key ↓ (cid) Primary Key ↓ (sid)

cid	grade	sid
CS101	CC	53831
CS203	BB	53832
CS112	AB	53650
HS105	BB	53666
⊥	BB	53832

Primary key - 13 (a) violation

Why NULL values become an issue

register

cid	grade	**sid**
CS101	CC	53831
CS203	BB	53832
CS112	AB	53650
HS105	BB	53666
$\perp$	BB	$\perp$
$\perp$	BB	$\perp$

Primary key - 13 (a) violation

Why NULL values become an issue

	register	
cid	grade	**sid**
CS101	CC	53831
CS203	BB	53832
CS112	AB	53650
HS105	BB	53666
$\perp$	BB	$\perp$
$\perp$	BB	$\perp$

Cannot disting two $\perp$ values. That is the test: $\perp == \perp$ will NOT evaluate to TRUE!

Primary key - 14 violation

None of the columns of the primary key should have ⊥ values

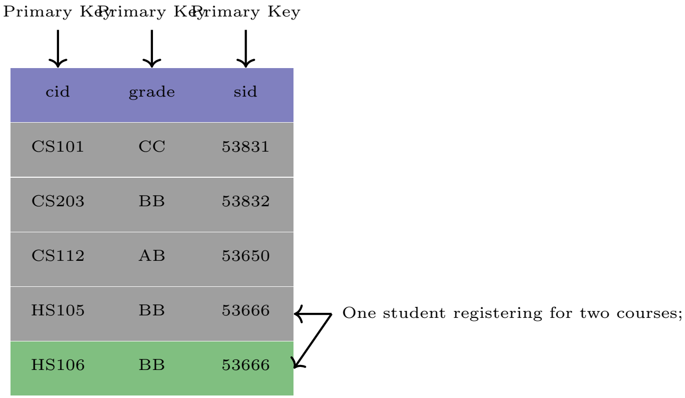
Primary Key ↓ cid

Primary Key ↓ sid

cid	grade	sid
CS101	CC	53831
CS203	BB	53832
CS112	AB	53650
HS105	BB	53666
HS105	BB	⊥

Primary key - 15 more than one column

Can these three columns put togather be primary key? - yes



Primary Key Primary Key Primary Key

cid	grade	sid
CS101	CC	53831
CS203	BB	53832
CS112	AB	53650
HS105	BB	53666
HS106	BB	53666

One student registering for two courses;

Primary Key - 16 more than one column

Observe for change in meaning

Primary Key Primary Key

cid	grade	sid
CS101	CC	53831
CS203	BB	53832
CS112	AB	53650
HS105	BB	53666
HS106	BB	53666

allow one student to register more than one course

- Allows two or more students to register for one course
- Allows one student to register for two or more courses

Primary key - 17 more than one column

Meaning of two columns to be the primary key

Primary Key ↓ cid

Primary Key ↓ sid

cid	grade	sid
CS101	CC	53831
CS203	BB	53832
CS112	AB	53650
HS105	BB	53666
HS105	BB	53667

allow one course to register more than one student

- Allows two or more students to register for one course
- Allows one student to register for two or more courses

Primary key - 18 more than one column

Observe for change in meaning

Primary Key

cid	grade	sid
CS101	CC	53831
CS203	BB	53832
CS112	AB	53650
HS105	BB	53666
HS106	BB	53666

Cannot allow one student to register for two different courses

Primary key - 19 more than one column

Observe for change in meaning

Primary Key

cid	grade	sid
CS101	CC	53831
CS203	BB	53832
CS112	AB	53650
HS105	BB	53666
HS105	BB	53667

Cannot allow multiple students to register for one course

Uniqueness

Primary Key ↓ (sid)   Uniqueness ↓ (login)

sid	name	login	age	spi
50000	Dave	dave	19	8.25
53666	Jones	jones	18	8.50
53688	Smith	smith	18	8.00
53650	Smith	smit	18	9.50
53831	Madayan	madayan	18	4.50
53832	Guldu	abcdu	18	5.00

Uniqueness Violation

Cannot have two rows having identical `login` values

Primary Key

Uniqueness

sid	name	login	age	spi
50000	Dave	dave	19	8.25
53666	Jones	jones	18	8.50
53688	Smith	smith	18	8.00
53650	Smith	smit	18	9.50
53831	Madayan	madayan	18	4.50
53832	Guldu	abcdu	18	5.00
53835	Dave D	dave	19	7.00

Not NULL - has implicit meaning - 01

Primary Key		Uniqueness		Not Null
sid	name	login	age	spi
50000	Dave	dave	19	8.25
53666	Jones	jones	18	8.50
53688	Smith	smith	18	8.00
53650	Smith	smit	18	9.50
53831	Madayan	madayan	18	4.50
53832	Guldu	abcdu	18	5.00

Not NULL - has implicit meaning - 02

Cannot have a cell taking ⊥ (NULL) value

Last row is illegal

Primary Key ↓ sid

Uniqueness ↓ login

Not Null ↓ spi

sid	name	login	age	spi
50000	Dave	dave	19	8.25
53666	Jones	jones	18	8.50
53688	Smith	smith	18	8.00
53650	Smith	smit	18	9.50
53831	Madayan	madayan	18	4.50
53832	Guldu	abcdu	18	5.00
53835	Atul	atulp	19	⊥

DEFAULT value

Primary Key

Uniqueness

Not Null

sid	name	login	age	spi
50000	Dave	dave	19	8.25
53666	Jones	jones	18	8.50
53688	Smith	smith	18	8.00
53650	Smith	smit	18	9.50
53831	Madayan	madayan	18	4.50
53832	Guldu	abcdu	18	5.00
53835	Atul	atulp	100	8.5

- Assume column `age` has DEFAULT value 100
- Assume you are inserting a new row
- You have specified values for `sid`, `name`, `login`, and `spi`
- Not specified any value for `age`
- The DEFAULT constraints is responsible to write value 100 to the `age` of the new row

Foreign Key - 01

Primary Key

↓

sid	name	login	age	spi
50000	Dave	dave	19	8.25
53666	Jones	jones	18	8.50
53688	Smith	smith	18	8.00
53650	Smith	smith	18	9.50
53831	Madayan	madayan	18	4.50
53832	Guldu	abcdu	18	5.00

Foreign Key - 02

cid	grade	studid
CS101	CC	53831
CS203	BB	53832
CS112	AB	53650
HS105	BB	53666

Primary Key

sid	name	login	age	spi
50000	Dave	dave	19	8.25
53666	Jones	jones	18	8.50
53688	Smith	smith	18	8.00
53650	Smith	smith	18	9.50
53831	Madayan	madayan	18	4.50
53832	Guldu	abcdu	18	5.00

Foreign Key - 03

Foreign Key

cid	grade	studid
CS101	CC	53831
CS203	BB	53832
CS112	AB	53650
HS105	BB	53666

Primary Key

sid	name	login	age	spi
50000	Dave	dave	19	8.25
53666	Jones	jones	18	8.50
53688	Smith	smith	18	8.00
53650	Smith	smith	18	9.50
53831	Madayan	madayan	18	4.50
53832	Guldu	abcdu	18	5.00

Foreign Key - 04

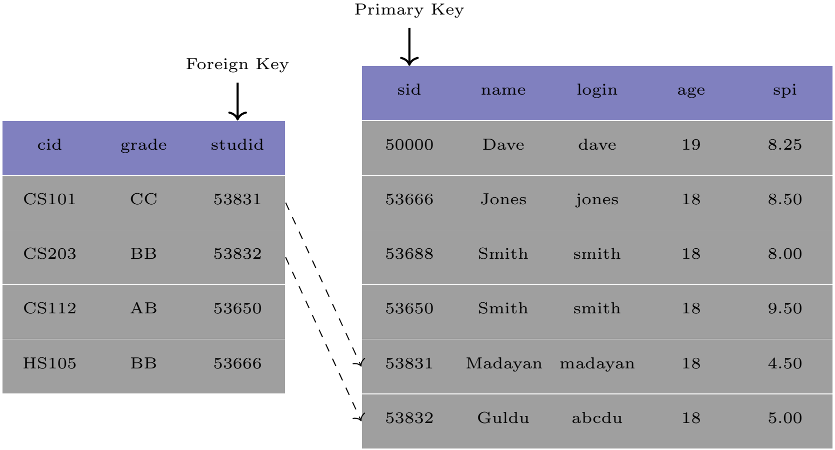
Foreign Key

cid	grade	studid
CS101	CC	53831
CS203	BB	53832
CS112	AB	53650
HS105	BB	53666

Primary Key

sid	name	login	age	spi
50000	Dave	dave	19	8.25
53666	Jones	jones	18	8.50
53688	Smith	smith	18	8.00
53650	Smith	smith	18	9.50
53831	Madayan	madayan	18	4.50
53832	Guldu	abcdu	18	5.00

Foreign Key - 05



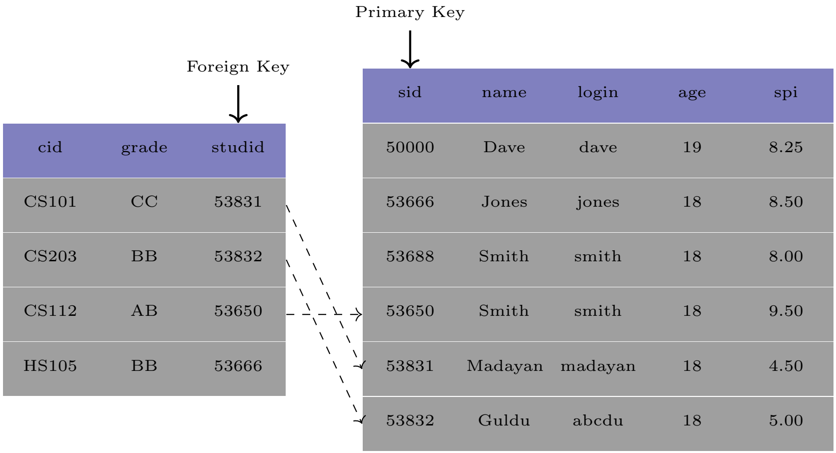
Foreign Key

cid	grade	studid
CS101	CC	53831
CS203	BB	53832
CS112	AB	53650
HS105	BB	53666

Primary Key

sid	name	login	age	spi
50000	Dave	dave	19	8.25
53666	Jones	jones	18	8.50
53688	Smith	smith	18	8.00
53650	Smith	smith	18	9.50
53831	Madayan	madayan	18	4.50
53832	Guldu	abcdu	18	5.00

Foreign Key - 06



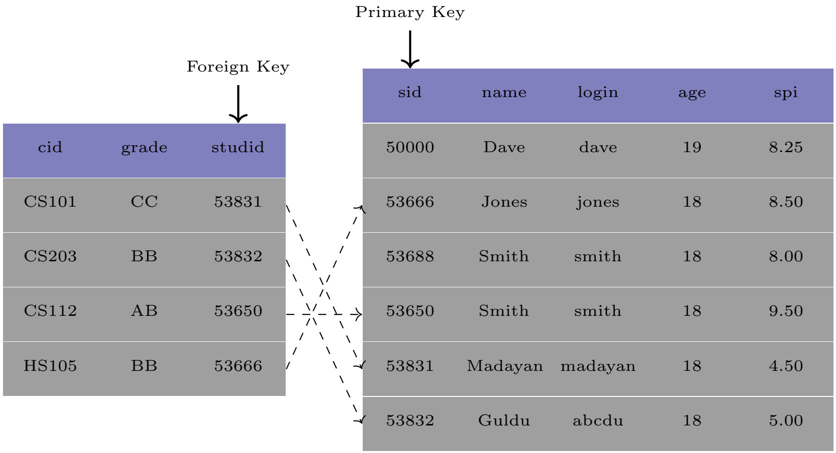
Primary Key

Foreign Key

cid	grade	studid
CS101	CC	53831
CS203	BB	53832
CS112	AB	53650
HS105	BB	53666

sid	name	login	age	spi
50000	Dave	dave	19	8.25
53666	Jones	jones	18	8.50
53688	Smith	smith	18	8.00
53650	Smith	smith	18	9.50
53831	Madayan	madayan	18	4.50
53832	Guldu	abcdu	18	5.00

Foreign Key - 07



Foreign Key

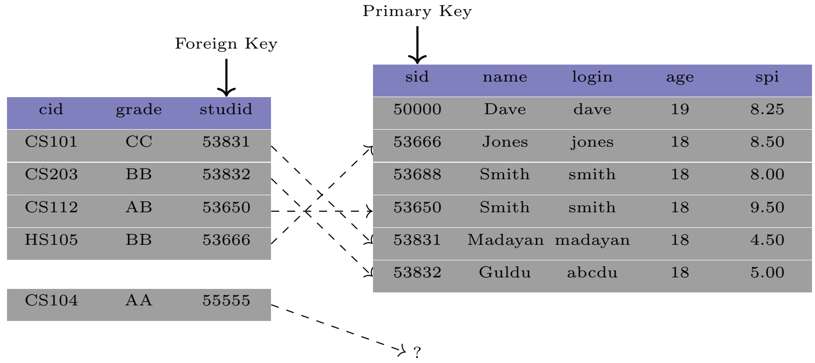
Primary Key

cid	grade	studid
CS101	CC	53831
CS203	BB	53832
CS112	AB	53650
HS105	BB	53666

sid	name	login	age	spi
50000	Dave	dave	19	8.25
53666	Jones	jones	18	8.50
53688	Smith	smith	18	8.00
53650	Smith	smith	18	9.50
53831	Madayan	madayan	18	4.50
53832	Guldu	abcdu	18	5.00

Foreign Key 08 (Referential Integrity)

Cannot insert `CS104` into LHS table as studid=55555 doesn't exist in RHS table `sid` column

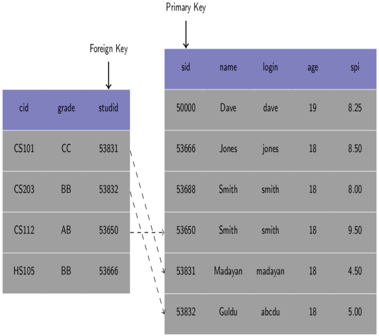


cid	grade	studid
CS101	CC	53831
CS203	BB	53832
CS112	AB	53650
HS105	BB	53666

CS104	AA	55555

sid	name	login	age	spi
50000	Dave	dave	19	8.25
53666	Jones	jones	18	8.50
53688	Smith	smith	18	8.00
53650	Smith	smith	18	9.50
53831	Madayan	madayan	18	4.50
53832	Guldu	abcdu	18	5.00

Foreign Key - Cases



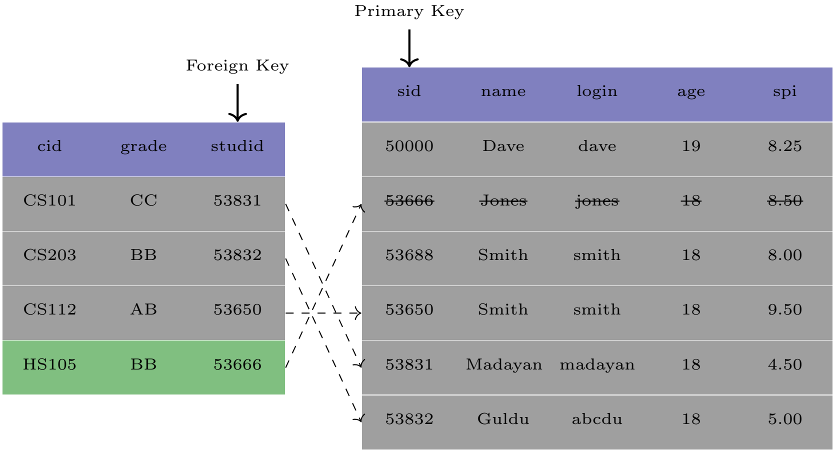
cid	grade	studid
CS101	CC	53831
CS203	BB	53832
CS112	AB	53650
HS105	BB	53666

sid	name	login	age	spi
50000	Dave	dave	19	8.25
53666	Jones	jones	18	8.50
53688	Smith	smith	18	8.00
53650	Smith	smith	18	9.50
53831	Madayan	madayan	18	4.50
53832	Guldu	abcdu	18	5.00

Operations on LHS & RHS tables

1. A row is deleted from RHS table
2. A row is updated in RHS table
3. A row is inserted into LHS table
4. A row is deleted from LHR table

Foreign Key - Row Deletion from RHS Table



Primary Key

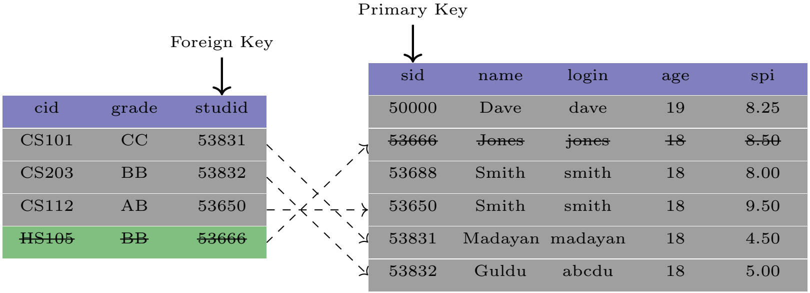
Foreign Key

cid	grade	studid
CS101	CC	53831
CS203	BB	53832
CS112	AB	53650
HS105	BB	53666

sid	name	login	age	spi
50000	Dave	dave	19	8.25
~~53666~~	~~Jones~~	~~jones~~	~~18~~	~~8.50~~
53688	Smith	smith	18	8.00
53650	Smith	smith	18	9.50
53831	Madayan	madayan	18	4.50
53832	Guldu	abcdu	18	5.00

Foreign Key - Row Deletion from RHS Table - Action 01

Delete all rows in LHS table with studid=53666



Foreign Key

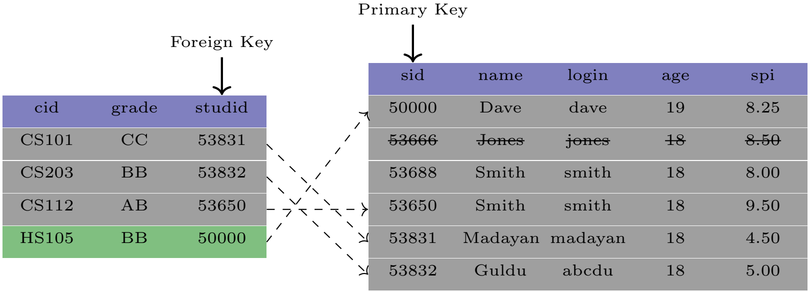
cid	grade	studid
CS101	CC	53831
CS203	BB	53832
CS112	AB	53650
~~HS105~~	~~BB~~	~~53666~~

Primary Key

sid	name	login	age	spi
50000	Dave	dave	19	8.25
~~53666~~	~~Jones~~	~~jones~~	~~18~~	~~8.50~~
53688	Smith	smith	18	8.00
53650	Smith	smith	18	9.50
53831	Madayan	madayan	18	4.50
53832	Guldu	abcdu	18	5.00

Foreign Key - Row Deletion from RHS Table - Action 02

Write a DEFAULT values (say 50000) in all rows in LHS table with studid=53666

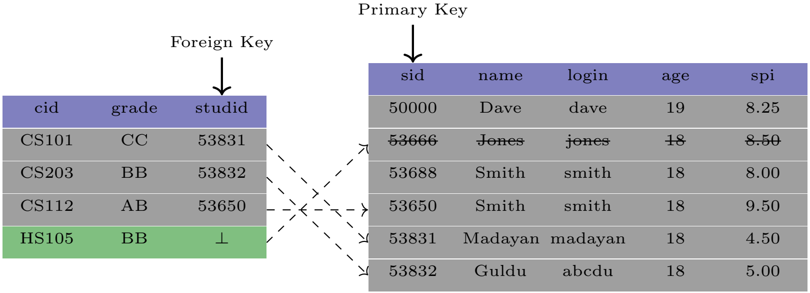


cid	grade	studid
CS101	CC	53831
CS203	BB	53832
CS112	AB	53650
HS105	BB	50000

sid	name	login	age	spi
50000	Dave	dave	19	8.25
~~53666~~	~~Jones~~	~~jones~~	~~18~~	~~8.50~~
53688	Smith	smith	18	8.00
53650	Smith	smith	18	9.50
53831	Madayan	madayan	18	4.50
53832	Guldu	abcdu	18	5.00

Foreign Key - Row Deletion from RHS Table - Action 03

Write a NULL values in all rows in LHS table with studid=53666



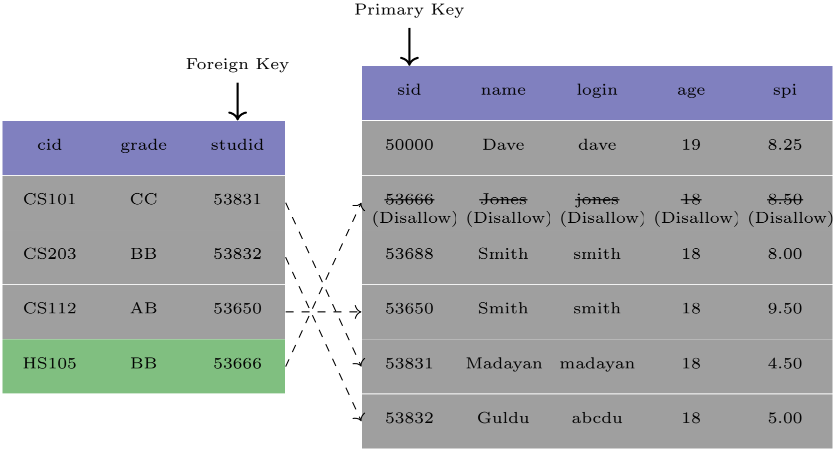
Primary Key

Foreign Key

cid	grade	studid
CS101	CC	53831
CS203	BB	53832
CS112	AB	53650
HS105	BB	⊥

sid	name	login	age	spi
50000	Dave	dave	19	8.25
~~53666~~	~~Jones~~	~~jones~~	~~18~~	~~8.50~~
53688	Smith	smith	18	8.00
53650	Smith	smith	18	9.50
53831	Madayan	madayan	18	4.50
53832	Guldu	abcdu	18	5.00

Foreign Key - Row Deletion from RHS Table - Action 04



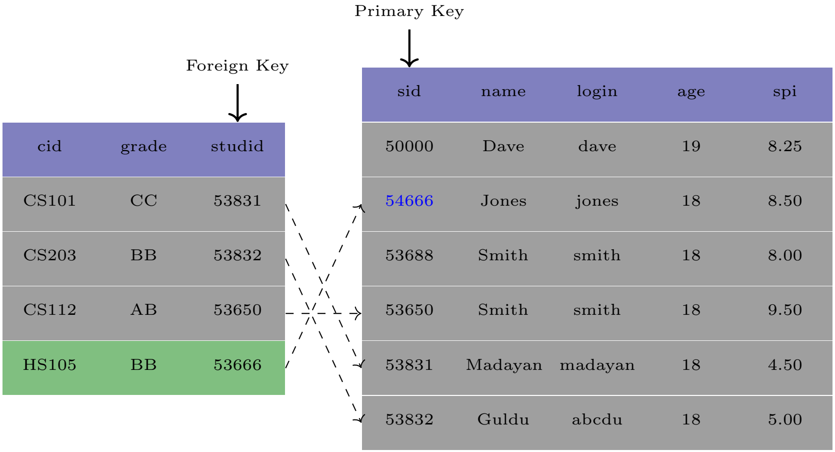
Foreign Key

cid	grade	studid
CS101	CC	53831
CS203	BB	53832
CS112	AB	53650
HS105	BB	53666

Primary Key

sid	name	login	age	spi
50000	Dave	dave	19	8.25
~~53666~~ (Disallow)	~~Jones~~ (Disallow)	~~jones~~ (Disallow)	~~18~~ (Disallow)	~~8.50~~ (Disallow)
53688	Smith	smith	18	8.00
53650	Smith	smith	18	9.50
53831	Madayan	madayan	18	4.50
53832	Guldu	abcdu	18	5.00

Foreign Key - Updated a row in RHS Table



Primary Key

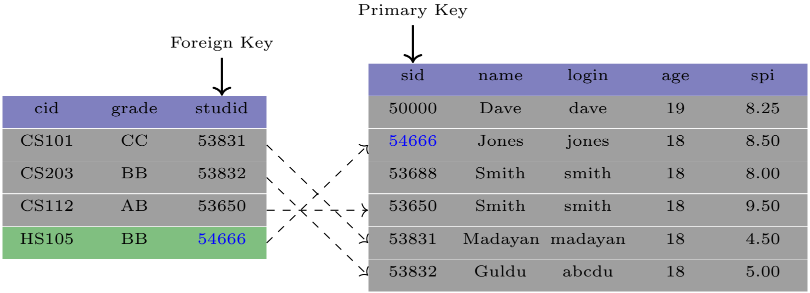
Foreign Key

cid	grade	studid
CS101	CC	53831
CS203	BB	53832
CS112	AB	53650
HS105	BB	53666

sid	name	login	age	spi
50000	Dave	dave	19	8.25
54666	Jones	jones	18	8.50
53688	Smith	smith	18	8.00
53650	Smith	smith	18	9.50
53831	Madayan	madayan	18	4.50
53832	Guldu	abcdu	18	5.00

Foreign Key - Row in RHS Table Updated - Action 01

Update all rows in LHS table with studid=54666



Foreign Key

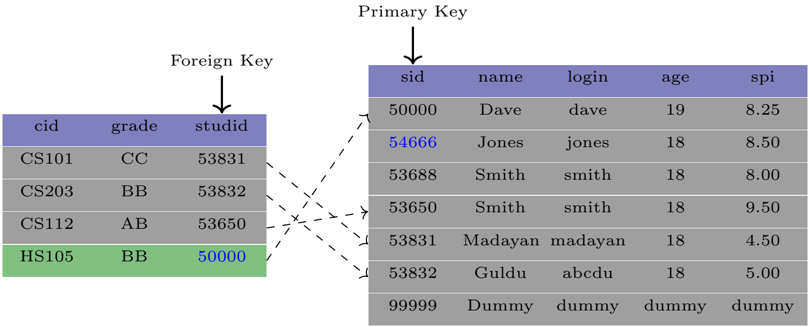
cid	grade	studid
CS101	CC	53831
CS203	BB	53832
CS112	AB	53650
HS105	BB	54666

Primary Key

sid	name	login	age	spi
50000	Dave	dave	19	8.25
54666	Jones	jones	18	8.50
53688	Smith	smith	18	8.00
53650	Smith	smith	18	9.50
53831	Madayan	madayan	18	4.50
53832	Guldu	abcdu	18	5.00

Foreign Key - Row in RHS Table Updated - Action 02

Write a DEFAULT values (say 50000) in all rows in LHS table with studid=53666



Primary Key

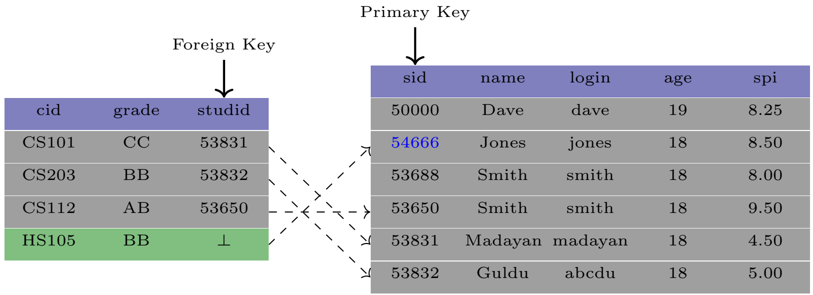
Foreign Key

cid	grade	studid
CS101	CC	53831
CS203	BB	53832
CS112	AB	53650
HS105	BB	50000

sid	name	login	age	spi
50000	Dave	dave	19	8.25
54666	Jones	jones	18	8.50
53688	Smith	smith	18	8.00
53650	Smith	smith	18	9.50
53831	Madayan	madayan	18	4.50
53832	Guldu	abcdu	18	5.00
99999	Dummy	dummy	dummy	dummy

Foreign Key - Row in RHS Table Updated - Action 03

Write a NULL values in all rows in LHS table with studid=53666



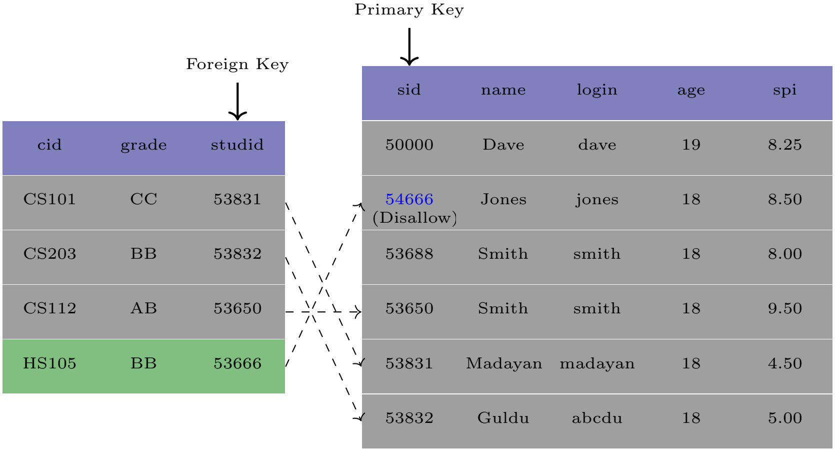
Foreign Key

cid	grade	studid
CS101	CC	53831
CS203	BB	53832
CS112	AB	53650
HS105	BB	⊥

Primary Key

sid	name	login	age	spi
50000	Dave	dave	19	8.25
54666	Jones	jones	18	8.50
53688	Smith	smith	18	8.00
53650	Smith	smith	18	9.50
53831	Madayan	madayan	18	4.50
53832	Guldu	abcdu	18	5.00

Foreign Key - Row in RHS Table Updated - Action 04



Primary Key

Foreign Key

cid	grade	studid
CS101	CC	53831
CS203	BB	53832
CS112	AB	53650
HS105	BB	53666

sid	name	login	age	spi
50000	Dave	dave	19	8.25
54666 (Disallow)	Jones	jones	18	8.50
53688	Smith	smith	18	8.00
53650	Smith	smith	18	9.50
53831	Madayan	madayan	18	4.50
53832	Guldu	abcdu	18	5.00

Table operations

Table Operations

Single Table

- Discussing operations on a single table
- Create table with specified number of columns, their names and their data types
- Delete table
- Add a column to the existing table (at the beginning)
- Add a column to the existing table (in the middle)
- Add a column to the existing table (at the end)
- Delete a column from an existing table
- Change column data type

Table Operations

Single Table

- Add a constraint to existing table
- Delete an existing constraint from a table

Brief history

Relational Databases

Brief History

- E. F. Codd (IBM Research Laboratory) invented the Relational Databases
- Was awarded Turing award in 1981 for the seminal work
- Try to read the paper A Relational Model of Data for Large Shared Data Banks
- A theoretical model defining relations, and operations on relations
- Followed by 12 rules of Codd for the relational databases

Relational Databases - Implementations

Brief History

- 1974: IBM's System R prototype of RDBMS
- 1979: Oracle Corporation's Oracle
- 1970: Ingres by University of California
- 1987: SAP by Sybase
- Try to read the paper A Relational Model of Data for Large Shared Data Banks
- A theoretical model defining relations, and operations on relations
- Followed by 12 rules of Codd for the relational databases

SQL Language

Brief History

- Structured Query Language (SQL)
- Designed for managing data in RDBMS
- first version is known as SEQUEL (Structured English Query Language)
- Developed by Donald D. Chamberlin and Raymond F. Boyce
- First standard is available in the year 1986 formalized by ANSI (SQL-86)
- Latest standard is published in 2019 (SQL:2019)

SQL

SQL

Overview

DDL Subset of SQL support creation, deletion and modification of tables and views

DML Subset of SQL that allows users to pose queries, insert, delete and modify tuples

Triggers, Events & Adv. Constraints Performs operations based on actions or time

Embedded SQL SQL statements can be included in various programming languages such as C, C++, Java, python and/or php

SQL

Overview

Transaction Management Various commands allow user to explicitly control aspects of how a transaction is to be executed

Security provide mechanism to control user's access to tables and views

Programming Constructs such as control statements, loops, exceptions, error handling statements