

SQL

Relational Databases

- From IBM ~ 1970s
- Data stored in Tabular format:
 - Columns of tables: fields (name, address, department, ...)
 - Rows of tables: individual entries (student1, student2, ...)
- Key: unique way of accessing a given row

Relational Databases

- From IBM ~ 1970s
- Data stored in Tabular format:
 - Columns of tables: fields (name, address, department, ...)
 - Rows of tables: individual entries (student1, student2, ...)
- Key: unique way of accessing a given row
 - **Primary key**: important for fast access on large databases

Relational Databases

- From IBM ~ 1970s
- Data stored in Tabular format:
 - Columns of tables: fields (name, address, department, ...)
 - Rows of tables: individual entries (student1, student2, ...)
- Key: unique way of accessing a given row
 - **Primary key**: important for fast access on large databases
 - **Foreign key**: connect to a different table - **Relationships**

Queries

- Retrieve data from the database:

eg. “Find students with name beginning with A”

“Find all courses offered in 2021”

Structured Query Language (SQL)

- English - like, but structured
- Quite verbose
- Specific mathematical operations:
 - Inner Join
 - Outer Join

Example: Inner Join

Name	IDNumber	hostelID
Sunil Shashi	MAD001	1
Chetana Anantha	MAD002	2
Madhur Prakash	MAD003	2
Nihal Surya	MAD004	3
Shweta Lalita	MAD005	2
Raghu Balwinder	MAD006	3
Gulshan Kuldeep	MAD007	1
Kishan Shrivatsa	MAD008	1
Purnima Sunil	MAD009	2
Nikitha Madhavi	MAD010	1
Lilavati Prabhakar	MAD011	3
Rama Yamuna	MAD012	3

ID	Name	Capacity
1	Jamuna	300
2	Ganga	300
3	Brahmaputra	500

Student - Hostel mapping

```
select Students.name, Hostels.name  
  from Students  
 inner join Hostels  
on Students.hostelID = Hostels.ID
```


Student - Hostel mapping

```
select Students.name, Hostels.name  
  from Students  
 inner join Hostels  
on Students.hostelID = Hostels.ID
```

Sunil Shashi, Jamuna

Chetana Anantha, Ganga

Cartesian Product

- N entries in table 1
- M entries in table 2
- M x N combinations - filter on them

Powerful SQL queries can be constructed

Example: find all students in Calculus

- Find ID number for course
- Look up StudentsCourses table to find all entries with this course ID
- Look up Students to find names of students with these IDs

Example: find all students in Calculus

- Find ID number for course
- Look up StudentsCourses table to find all entries with this course ID
- Look up Students to find names of students with these IDs

```
SELECT s.name  
FROM Students s
```

Example: find all students in Calculus

- Find ID number for course
- Look up StudentsCourses table to find all entries with this course ID
- Look up Students to find names of students with these IDs

```
SELECT s.name  
FROM Students s  
JOIN StudentsCourses sc ON s.IDNumber = sc.studentID
```

Example: find all students in Calculus

- Find ID number for course
- Look up StudentsCourses table to find all entries with this course ID
- Look up Students to find names of students with these IDs

```
SELECT s.name  
FROM Students s  
JOIN StudentsCourses sc ON s.IDNumber = sc.studentID  
JOIN Courses c ON c.ID = sc.courseID
```

Example: find all students in Calculus

- Find ID number for course
- Look up StudentsCourses table to find all entries with this course ID
- Look up Students to find names of students with these IDs

```
SELECT s.name
FROM Students s
JOIN StudentsCourses sc ON s.IDNumber = sc.studentID
JOIN Courses c ON c.ID = sc.courseID
WHERE c.name='Calculus'
```

Summary

- Models - persistent data storage
- Mechanisms:
 - CSV, Spreadsheets, SQL, NoSQL
- Entities and Relationships
 - Different ways of representing

No details on display, views, or what kind of updates permitted