Introduction to Machine Learning

SQL and Analytics

Andres Mendez-Vazquez

January 9, 2023

Outline

Outline

Introduction [1]

Observation

- Collecting and storing data for analysis is a very human activity.

For this we have Data Analysis

- Many different names are used to describe the discipline of data analysis.

Introduction [1]

Observation

- Collecting and storing data for analysis is a very human activity.

For this we have Data Analysis

- Many different names are used to describe the discipline of data analysis.

Definition

A somewhat Definition

- Data analysis is part data discovery, part data interpretation, and part data communication.

It Purpose

- To improve decision making
  - By humans
  - By machines through automation

Properties

- It requires not only sound methodology, but also curiosity - the **Why?**

Definition

A somewhat Definition

- Data analysis is part data discovery, part data interpretation, and part data communication.

It Purpose

- To improve decision making
  - By humans
  - By machines through automation

Properties

- It requires not only sound methodology, but also curiosity - the **Why?**

Definition

A somewhat Definition

- Data analysis is part data discovery, part data interpretation, and part data communication.

It Purpose

- To improve decision making
  - By humans
  - By machines through automation

Properties

- It requires not only sound methodology, but also curiosity - the **Why?**

Be Careful About Data Analysis

It is basically an attempt to bring Statistics into CS

- Actually Statistics became part of Machine Learning as Data Science became widely accepted.

Definition

- Data Analysis is the application of Statistics in a Computer Science Framework

Basically

- As you can imagine they used SQL for extracting the samples for the experiments

Be Careful About Data Analysis

It is basically an attempt to bring Statistics into CS

- Actually Statistics became part of Machine Learning as Data Science became widely accepted.

Definition

- Data Analysis is the application of Statistics in a Computer Science Framework

Basically

- As you can imagine they used SQL for extracting the samples for the experiments

Be Careful About Data Analysis

It is basically an attempt to bring Statistics into CS

- Actually Statistics became part of Machine Learning as Data Science became widely accepted.

Definition

- Data Analysis is the application of Statistics in a Computer Science Framework

Basically

- As you can imagine they used SQL for extracting the samples for the experiments

Cautionary Tale

Observation

- Data analysis is by definition done on historical data.

The Present

- It does not predict the future.

Actually

- Criticisms are leveled against data analysis for being backward looking.
  - But many organizations are gaining knowledge of their process using it

Cautionary Tale

Observation

- Data analysis is by definition done on historical data.

The Present

- It does not predict the future.

Actually

- Criticisms are leveled against data analysis for being backward looking.
  - But many organizations are gaining knowledge of their process using it

Cautionary Tale

Observation

- Data analysis is by definition done on historical data.

The Present

- It does not predict the future.

Actually

- Criticisms are leveled against data analysis for being backward looking.
  - But many organizations are gaining knowledge of their process using it

Outline

SQL and Analytic

Actually [2]

- SQL is the language used to communicate with databases.

It is a general purpose language?

- It is not a general purpose language in the way that C or Python are.

But it is powerful enough

- SQL can help you get the job of data analysis done.

SQL and Analytic

Actually [2]

- SQL is the language used to communicate with databases.

It is a general purpose language?

- It is not a general purpose language in the way that C or Python are.

But it is powerful enough

- SQL can help you get the job of data analysis done.

SQL and Analytic

Actually [2]

- SQL is the language used to communicate with databases.

It is a general purpose language?

- It is not a general purpose language in the way that C or Python are.

But it is powerful enough

- SQL can help you get the job of data analysis done.

A Little History

IBM was the first to develop SQL databases

- From the relational model invented by Edgar Codd in 1969
  - A DARPA project

Something Notable

- From the beginning, there has been tension between computer theory and commercial reality.

But we need to look a little bit to the Algebra pushing for SQL

- Yes Relational Algebra

A Little History

IBM was the first to develop SQL databases

- From the relational model invented by Edgar Codd in 1969
  - A DARPA project

Something Notable

- From the beginning, there has been tension between computer theory and commercial reality.

But we need to look a little bit to the Algebra pushing for SQL

- Yes Relational Algebra

A Little History

IBM was the first to develop SQL databases

- From the relational model invented by Edgar Codd in 1969
  - A DARPA project

Something Notable

- From the beginning, there has been tension between computer theory and commercial reality.

But we need to look a little bit to the Algebra pushing for SQL

- Yes Relational Algebra

Outline

Here, Relational Algebra

The Relational Model

- Simple and uniform data structures – relations – and solid theoretical foundation
  - Which is important for query processing and optimization

Something Notable

- Relational Model is basis for most Database Management System (DBMS):
  - Oracle, Microsoft SQL Server, IBM DB2, Sybase, PostgreSQL, MySQL, Mariadb.

Typically used in conceptual design

- Either directly (creating tables using SQL DDL) or derived from a given **Entity-Relationship schema**.

Here, Relational Algebra

The Relational Model

- Simple and uniform data structures – relations – and solid theoretical foundation
  - Which is important for query processing and optimization

Something Notable

- Relational Model is basis for most Database Management System (DBMS):
  - Oracle, Microsoft SQL Server, IBM DB2, Sybase, PostgreSQL, MySQL, Mariadb.

Typically used in conceptual design

- Either directly (creating tables using SQL DDL) or derived from a given **Entity-Relationship schema**.

Here, Relational Algebra

The Relational Model

- Simple and uniform data structures – relations – and solid theoretical foundation
  - Which is important for query processing and optimization

Something Notable

- Relational Model is basis for most Database Management System (DBMS):
  - Oracle, Microsoft SQL Server, IBM DB2, Sybase, PostgreSQL, MySQL, Mariadb.

Typically used in conceptual design

- Either directly (creating tables using SQL DDL) or derived from a given **Entity-Relationship schema**.

Definition of a Relation

Definition

- A relation r over collection of sets (domain values)

$$D_1, D_2, ..., D_n \subseteq D_1 \times D_2 \times \cdots \times D_n$$

A relation thus is a set of n-tuples $(d_1, d_2, ..., d_n)$ where $d_i \in D_i$.

For example

- Given the sets

$$\begin{aligned} \text{StudentId} &= \{412, 307, 540\} \\ \text{StudentName} &= \{Smith, Jones\} \\ \text{Major} &= \{CS, CSE, BIO\} \end{aligned}$$

then $r = \{(412, Smith, CS), (307, Jones, CSE)\} \subseteq$ StudentId $\times$ StudentName $\times$ Major

Definition of a Relation

Definition

- A relation r over collection of sets (domain values)

$$D_1, D_2, ..., D_n \subseteq D_1 \times D_2 \times \cdots \times D_n$$

A relation thus is a set of n-tuples $(d_1, d_2, ..., d_n)$ where $d_i \in D_i$.

For example

- Given the sets

$$\begin{array}{cc} \text{StudentId} & = \{412, 307, 540\} \\ \text{StudentName} & \{Smith, Jones\} \\ \text{Major} & \{CS, CSE, BIO\} \end{array}$$

then $r = \{(412, Smith, CS), (307, Jones, CSE)\} \subseteq$ StudentId $\times$ StudentName $\times$ Major

Outline

Relation Schema

Definition

- Let $A_1, A_2, ..., A_n$ be attribute names with associated domains $D_1, D_2, ..., D_n$ then

$$R\,(A_1 : D_1, A_2 : D_2, ..., A_n : D_n)$$

is a relation schema. For example,
$Student$ (StudentId:integer, StudName:string, Major:string)

Properties

- A relation schema specifies the name and the structure of the relation.
- A collection of relation schemas is called a **relational database schema**.

Relation Schema

Definition

- Let $A_1, A_2, ..., A_n$ be attribute names with associated domains $D_1, D_2, ..., D_n$ then

$$R\,(A_1 : D_1, A_2 : D_2, ..., A_n : D_n)$$

is a relation schema. For example,
$Student$ (StudentId:integer, StudName:string, Major:string)

Properties

- A relation schema specifies the name and the structure of the relation.
- A collection of relation schemas is called a **relational database schema**.

Relation Instance

Definition

- A relation instance $r(R)$ of a relation schema can be thought of as a table with n columns and a number of rows.
  - Instead of relation instance we often just say relation.

Elements

- An element $t \in r(R)$ is called a tuple (or row).

Student	StudentId	StudentName	Major	←—Relation Schema
	412	Smith	CS	
	307	Jones	CSE	←—Tuple
	412	Smith	CSE	

Properties

- The order of rows is irrelevant
- There are no duplicate rows in a relation

Relation Instance

Definition

- A relation instance $r(R)$ of a relation schema can be thought of as a table with n columns and a number of rows.
  - Instead of relation instance we often just say relation.

Elements

- An element $t \in r(R)$ is called a tuple (or row).

Student	StudentId	StudentName	Major	⟵Relation Schema
	412	Smith	CS	
	307	Jones	CSE	⟵Tuple
	412	Smith	CSE	

Properties

- The order of rows is irrelevant
- There are no duplicate rows in a relation

Relation Instance

Definition

- A relation instance $r(R)$ of a relation schema can be thought of as a table with n columns and a number of rows.
  - Instead of relation instance we often just say relation.

Elements

- An element $t \in r(R)$ is called a tuple (or row).

Student	StudentId	StudentName	Major	⟵Relation Schema
	412	Smith	CS	
	307	Jones	CSE	⟵Tuple
	412	Smith	CSE	

Properties

- The order of rows is irrelevant
- There are no duplicate rows in a relation

Integrity Constraints in the Relational Model

Integrity Constraints (IC)

- It must be true for any instance of a relation schema (admissible instances)

Properties

- ICs are specified when the schema is defined.
- ICs are checked by the DBMS when relations (instances) are modified

Important

- If DBMS checks ICs, then the data managed by the DBMS more closely correspond to the real-world scenario that is being modeled!

Integrity Constraints in the Relational Model

Integrity Constraints (IC)

- It must be true for any instance of a relation schema (admissible instances)

Properties

- ICs are specified when the schema is defined.
- ICs are checked by the DBMS when relations (instances) are modified

Important

- If DBMS checks ICs, then the data managed by the DBMS more closely correspond to the real-world scenario that is being modeled!

Integrity Constraints in the Relational Model

Integrity Constraints (IC)

- It must be true for any instance of a relation schema (admissible instances)

Properties

- ICs are specified when the schema is defined.
- ICs are checked by the DBMS when relations (instances) are modified

Important

- If DBMS checks ICs, then the data managed by the DBMS more closely correspond to the real-world scenario that is being modeled!

Primary Key Constraints

A set of attributes is a **key** for a relation if

- No two distinct tuples have the same values for all key attributes.
  - This is not true for any subset of that key.

If there is more than one key for a relation

- We have a set of candidate keys then one is chosen (Data Base Administrator) to be the **primary key**.
  - Student(**StudId** : number, StudName : string, Major : string)
- For candidate keys not chosen as primary key, **uniqueness constraints** can be specified.
  - Note that it is often useful to introduce an artificial primary key - Actually at the Indexing

Primary Key Constraints

A set of attributes is a **key** for a relation if

- No two distinct tuples have the same values for all key attributes.
  - This is not true for any subset of that key.

If there is more than one key for a relation

- We have a set of candidate keys then one is chosen (Data Base Administrator) to be the **primary key**.
  - Student(**StudId** : number, StudName : string, Major : string)
- For candidate keys not chosen as primary key, **uniqueness constraints** can be specified.
  - Note that it is often useful to introduce an artificial primary key - Actually at the Indexing

Outline

Indexing

Something Notable

- A database index is a data structure that improves the speed of data retrieval operations on a database table.

This has costs

- At the cost of additional writes and storage space to maintain the index data structure.

For Example

- We can use B-Trees for indexing

Indexing

Something Notable

- A database index is a data structure that improves the speed of data retrieval operations on a database table.

This has costs

- At the cost of additional writes and storage space to maintain the index data structure.

For Example

- We can use B-Trees for indexing

Indexing

Something Notable

- A database index is a data structure that improves the speed of data retrieval operations on a database table.

This has costs

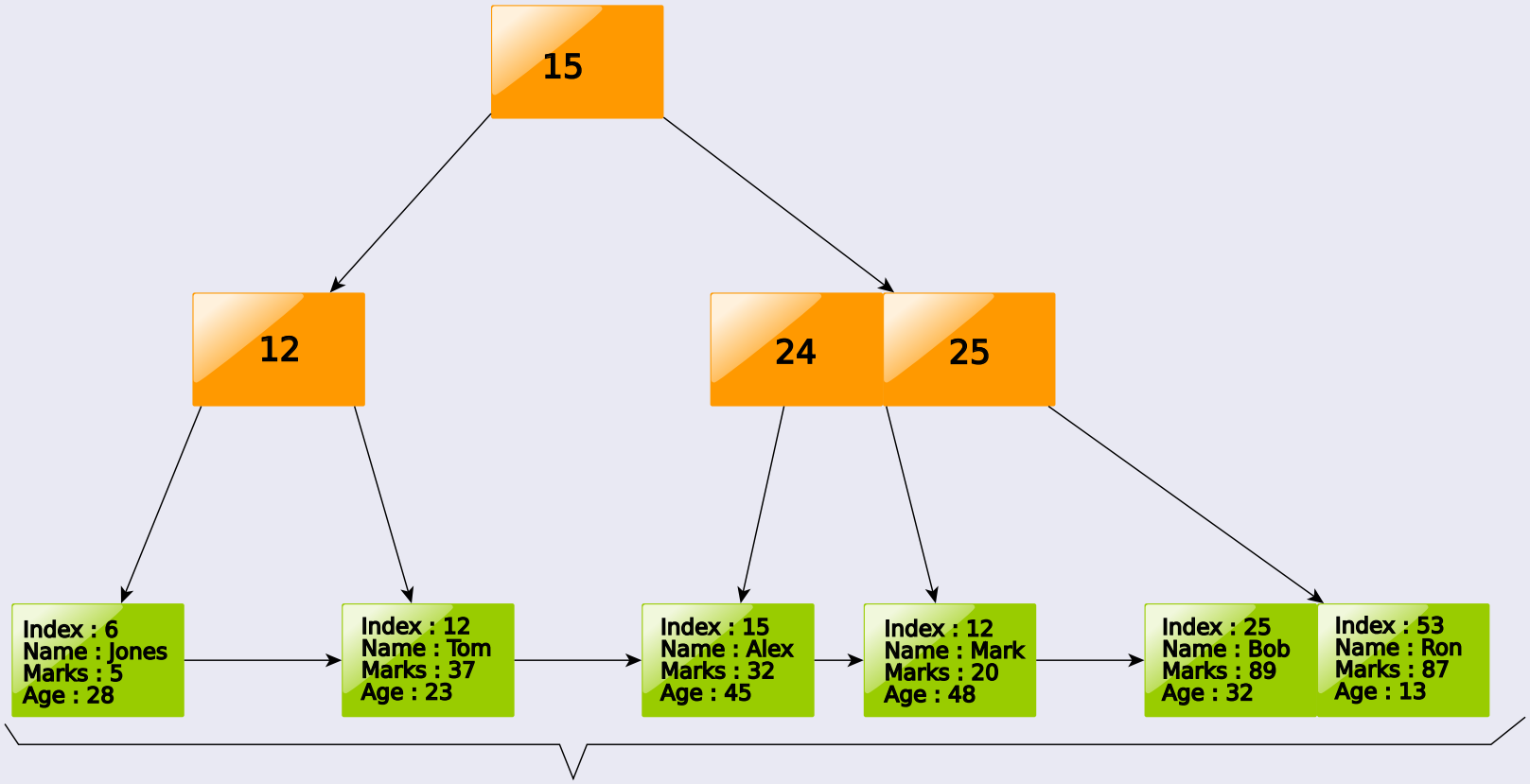
- At the cost of additional writes and storage space to maintain the index data structure.

For Example

- We can use B-Trees for indexing

B-Trees

Example



Index	Name	Marks	Age
6	Jones	5	28
12	Tom	37	23
⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮
53	Ron	87	13

We have the following complexities

Complexity

Type	Insertion	Deletion	Search
Unsorted Array	$O(1)$	$O(n)$	$O(n)$
Sorted Array	$O(n)$	$O(n)$	$O(\log n)$
B-Tree	$O(\log n)$	$O(\log n)$	$O(\log n)$

There are many other techniques

- Even the use of table clustering

We have the following complexities

Complexity

Type	Insertion	Deletion	Search
Unsorted Array	$O(1)$	$O(n)$	$O(n)$
Sorted Array	$O(n)$	$O(n)$	$O(\log n)$
B-Tree	$O(\log n)$	$O(\log n)$	$O(\log n)$

There are many other techniques

- Even the use of table clustering

Outline

As always constraints

We have the following

- Set of attributes in one relation (**child relation**) that is used to "refer" to a tuple in another relation (**parent relation**).
- Foreign key must refer to the primary key of the referenced relation.

Something Notable

- Foreign key attributes are required in relation schemas that have been derived from relationship types.
  - offers($\underbrace{(Prodname)}_{Primary\ Key} \rightarrow$ PRODUCTS, $\underbrace{(SName)}_{Primary\ Key} \rightarrow$ SUPPLIERS, Price)
  - orders($\underbrace{(FName, LName)}_{Primary\ Keys} \rightarrow$ CUSTOMERS, $\underbrace{(SName)}_{Foreign\ Key} \rightarrow$ SUPPLIERS, $\underbrace{(Prodname)}_{Foreign\ Key} \rightarrow$ PRODUCTS, Quantity)
- Foreign/primary key attributes must have matching domains.

As always constraints

We have the following

- Set of attributes in one relation (**child relation**) that is used to "refer" to a tuple in another relation (**parent relation**).
- Foreign key must refer to the primary key of the referenced relation.

Something Notable

- Foreign key attributes are required in relation schemas that have been derived from relationship types.
  - offers($\underbrace{(\boldsymbol{Prodname})}_{Primary\ Key} \rightarrow$ PRODUCTS, $\underbrace{(\boldsymbol{SName})}_{Primary\ Key} \rightarrow$ SUPPLIERS, Price)
  - orders($\underbrace{(FName, LName)}_{Primary\ Keys} \rightarrow$ CUSTOMERS, $\underbrace{(\boldsymbol{SName})}_{Foreign\ Key} \rightarrow$ SUPPLIERS, $\underbrace{(\boldsymbol{Prodname})}_{Foreign\ Key} \rightarrow$ PRODUCTS, Quantity)

  Foreign/primary key attributes must have matching domains.

Furthermore

A foreign key constraint is satisfied for a tuple if either

- Some values of the foreign key attributes are null (meaning a reference is not known)
- The values of the foreign key attributes occur as the values of the primary key (of some tuple) in the parent relation.

Something Notable

- The combination of foreign key attributes in a relation schema typically builds the primary key of the relation,
  - offers($\underbrace{(Prodname)}_{Primary\ Key} \rightarrow$ PRODUCTS, $\underbrace{(SName)}_{Primary\ Key} \rightarrow$ SUPPLIERS, Price)

Properties

- If all foreign key constraints are enforced for a relation, referential integrity is achieved, i.e., there are no dangling references.

Furthermore

A foreign key constraint is satisfied for a tuple if either

- Some values of the foreign key attributes are null (meaning a reference is not known)
- The values of the foreign key attributes occur as the values of the primary key (of some tuple) in the parent relation.

Something Notable

- The combination of foreign key attributes in a relation schema typically builds the primary key of the relation,
  - offers($\underbrace{(\boldsymbol{Prodname})}_{Primary\ Key} \rightarrow$ PRODUCTS, $\underbrace{(\boldsymbol{SName})}_{Primary\ Key} \rightarrow$ SUPPLIERS, Price)

Properties

- If all foreign key constraints are enforced for a relation, referential integrity is achieved, i.e., there are no dangling references.

Furthermore

A foreign key constraint is satisfied for a tuple if either

- Some values of the foreign key attributes are null (meaning a reference is not known)
- The values of the foreign key attributes occur as the values of the primary key (of some tuple) in the parent relation.

Something Notable

- The combination of foreign key attributes in a relation schema typically builds the primary key of the relation,
  - offers($\underbrace{(\boldsymbol{Prodname})}_{Primary\ Key} \rightarrow$ PRODUCTS, $\underbrace{(\boldsymbol{SName})}_{Primary\ Key} \rightarrow$ SUPPLIERS, Price)

Properties

- If all foreign key constraints are enforced for a relation, referential integrity is achieved, i.e., there are no dangling references.

Outline

Query Language

Database Manipulation

- A Query Language (QL) is a language that allows users to manipulate and retrieve data from a database.
- The relational model supports simple, powerful QLs.

Query Language $\not\vdash$ Programming Language

- SQL is not expected to be Turing Complete

Two (mathematical) Query Languages are the basis of modern SQL

- Relational Algebra: procedural, very useful for representing query execution plans, and query optimization techniques.
- Relational Calculus: declarative, logic based language

Query Language

Database Manipulation

- A Query Language (QL) is a language that allows users to manipulate and retrieve data from a database.
- The relational model supports simple, powerful QLs.

Query Language != Programming Language

- SQL is not expected to be Turing Complete

Two (mathematical) Query Languages are the basis of modern SQL

- Relational Algebra: procedural, very useful for representing query execution plans, and query optimization techniques.
- Relational Calculus: declarative, logic based language

Query Language

Database Manipulation

- A Query Language (QL) is a language that allows users to manipulate and retrieve data from a database.
- The relational model supports simple, powerful QLs.

Query Language != Programming Language

- SQL is not expected to be Turing Complete

Two (mathematical) Query Languages are the basis of modern SQL

- Relational Algebra: procedural, very useful for representing query execution plans, and query optimization techniques.
- Relational Calculus: declarative, logic based language

Relational Algebra

Six basic operators in relational algebra

Operation	Symbol	Description
Select	σ	selects a subset of tuples
Project	π	deletes unwanted columns
Cartesian Product	$\times$	allows to combine two relations
Set Difference	$-$	tuples in first relation but not from the second
Union	$\cup$	Union of two relations
Rename	ρ	renames attribute(s) and relation

Outline

Select Operation

Notation, $\sigma_P(r)$

$$\sigma_P(r) = \{t | t \in t \text{ and } P(t)\}$$

Something Notable

- P is a formula in propositional calculus, composed of conditions of the form

Select Operation

Notation, $\sigma_P(r)$

$$\sigma_P(r) = \{t | t \in t \text{ and } P(t)\}$$

Something Notable

- P is a formula in propositional calculus, composed of conditions of the form
  - $(\neq, =, <, > etc))$ <attribute> $(\wedge, \vee, \neg)$ <constant>

Example

Given the relation r

A	B	C	D
α	α	1	7
α	β	5	7
β	β	12	3
β	β	23	10

Then $\sigma_{(A=B) \wedge D > 5}$

A	B	C	D
α	α	1	7
β	β	23	10

Example

Given the relation r

A	B	C	D
α	α	1	7
α	β	5	7
β	β	12	3
β	β	23	10

Then $\sigma_{(A=B)\wedge D>5}$

A	B	C	D
α	α	1	7
β	β	23	10

Outline

Project Operation

Notation

$$\pi_{A_1,A_2,...,A_k}$$

- Where $A_1, A_2, ..., A_k$ are attributes names and r is a relation.

Something Notable

- The result of the projection operation is defined as the relation that has k columns obtained by erasing all columns from r that are not listed.
- Duplicate rows are removed from result because relations are sets.

Project Operation

Notation

$$\pi_{A_1,A_2,...,A_k}$$

- Where $A_1, A_2, ..., A_k$ are attributes names and r is a relation.

Something Notable

- The result of the projection operation is defined as the relation that has k columns obtained by erasing all columns from r that are not listed.
- Duplicate rows are removed from result because relations are sets.

Example

$\pi_{A,C}(r)$

A	B	C	D
α	α	1	7
α	β	1	7
α	β	1	3
β	β	23	10

$\Rightarrow$

A	C
α	1
β	23

Outline

Cartesian Product

Notation

$$r \times s = \{tq | t \in r \text{ and } q \in s\}$$

- Assume that attributes of $r(R)$ and $s(S)$ are disjoint, $R \cap S = \emptyset$.
- If not renaming needs to be applied

We have that

A	B
α	1
β	2

$\times$

C	D
α	10
β	25

$=$

A	B	C	D
α	1	α	10
α	1	β	25
β	2	α	10
β	2	β	25

Cartesian Product

Notation

$$r \times s = \{tq | t \in r \text{ and } q \in s\}$$

- Assume that attributes of $r(R)$ and $s(S)$ are disjoint, $R \cap S = \emptyset$.
- If not renaming needs to be applied

We have that

A	B
α	1
β	2

$\times$

C	D
α	10
β	25

$=$

A	B	C	D
α	1	α	10
α	1	β	25
β	2	α	10
β	2	β	25

Outline

Set Difference Operator

Notation: $r - s$ where both r and s are relations

$$r - s = \{t | t \in r \text{ and } t \notin s\}$$

For $r - s$ to be applicable

- r and s must have the same arity
- Attribute domains must be compatible

Set Difference Operator

Notation: $r - s$ where both r and s are relations

$$r - s = \{t | t \in r \text{ and } t \notin s\}$$

For $r - s$ to be applicable

- r and s must have the same arity
- Attribute domains must be compatible

Example

Something Notable

A	B
α	1
α	2
β	1

$-$

A	B
α	2
β	3

$=$

A	B
α	1
β	1

Please

Take a look to the other operators

- They are used in SQL


R. L. Ott and M. T. Longnecker, *An introduction to statistical methods and data analysis*.
Cengage Learning, 2015.


R. Elmasri, S. B. Navathe, R. Elmasri, and S. Navathe, *Fundamentals of Database Systems</Title*.
Springer, 2000.