COMP283-Lecture 5
Applied Database Management

Introduction	
Database Administration	Optimisation
	Normalisation

Comp283-Lecture 5

Optimisation: DB Keys

- Unique Constraints
  - Enforces uniqueness on the set of attributes defined in them.
  - Only one row can contain null
- Primary Keys
  - special case of a unique constraint
  - one or more columns that together uniquely define the record
  - table can only have one primary key
  - usually (and should be) indexed
  - Every table should have a primary key

Comp283-Lecture 5

Optimisation: DB Keys

- Foreign Key
  - When a relationship exists between two tables based on some common set of attributes, the integrity of the data can be enforced by defining a Foreign Key.
  - There is one primary table and one secondary.
  - Set of attributes in secondary referencing a key in the primary table
  - Attributes and structure of linked-to-key in Primary cannot be modified (though data can of course)
- Composite Key
  - a unique constraint made up of multiple columns.

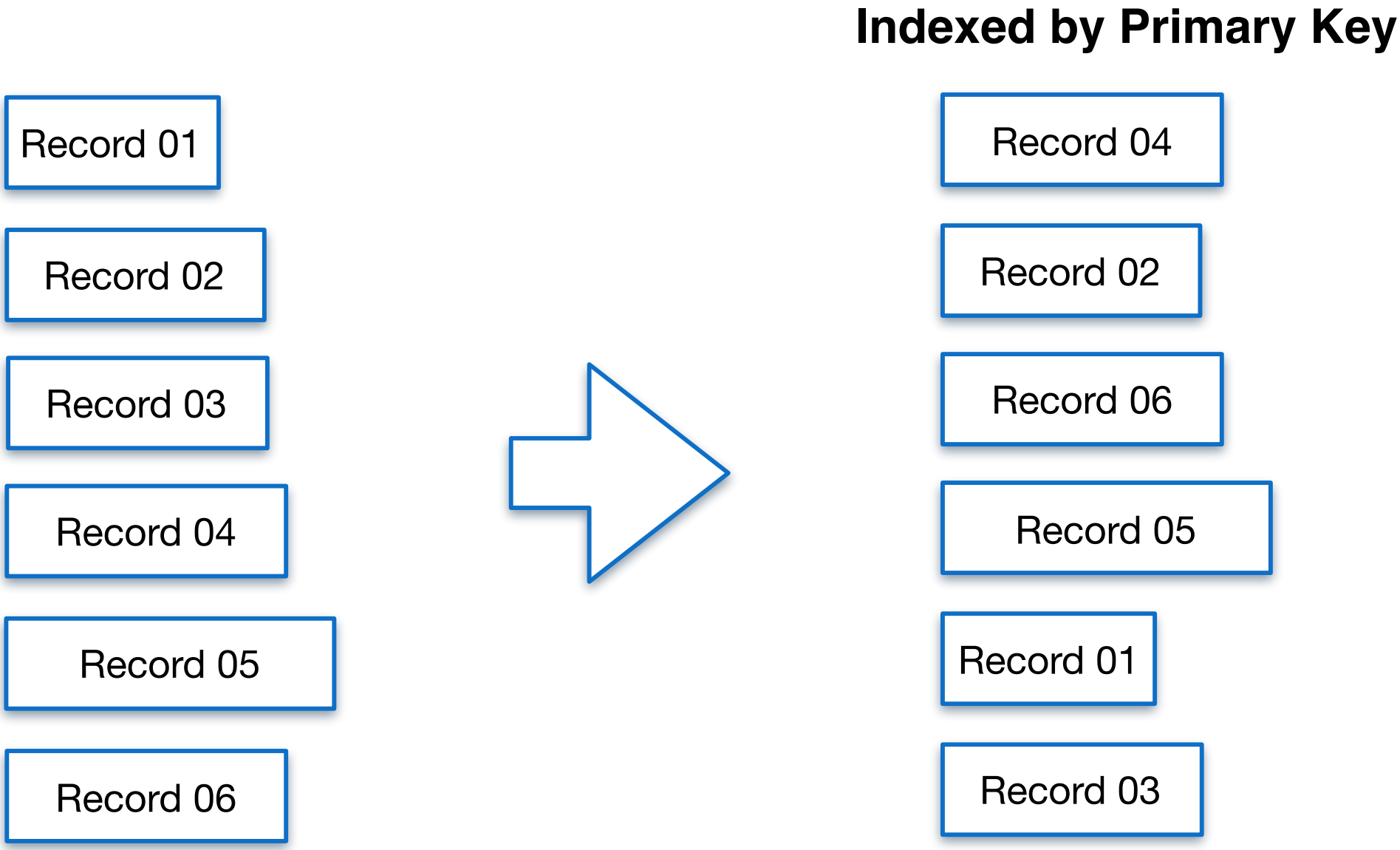
Comp283-Lecture 5

Optimisation : DB Indexes

- Clustered Indexes
  - Each table can have a maximum of 1 clustered index
  - Defines how the database will sort the table's data.
  - A clustered index is a stored (on-disk) structure associated with a specific table or view, designed to help speed up data retrieval.
  - Automatically created when a primary key is defined. Since every table should have a primary key, every table will have a clustered index.
- Un-Clustered Indexes
  - Additional indexes (if you choose to make them) are unclustered.

Comp283-Lecture 5

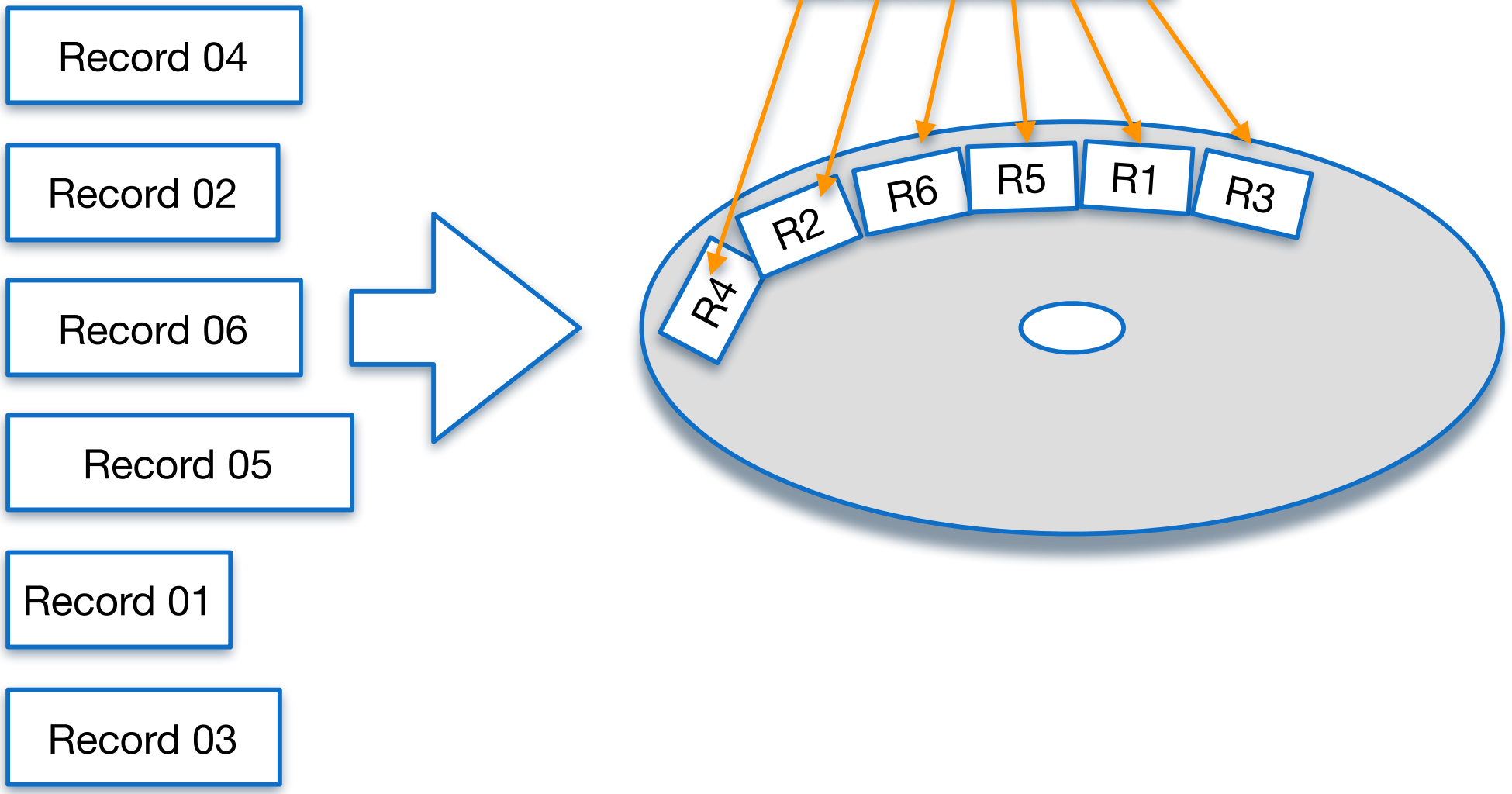
Optimisation : DB Indexes: Clustered Indexes



Comp283-Lecture 5

Optimisation : DB Indexes: Clustered Indexes

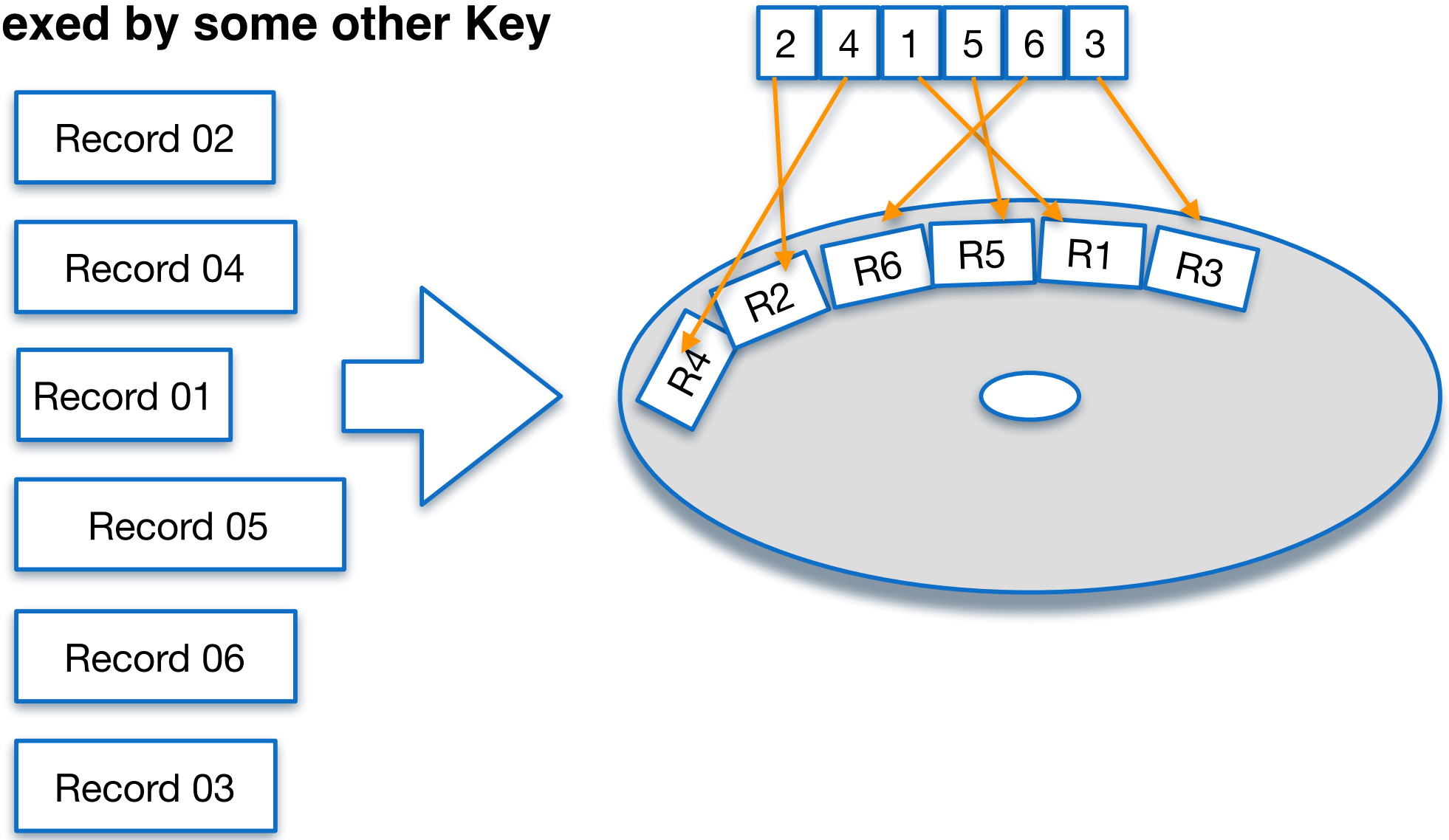
Indexed by Primary Key



Comp283-Lecture 5

Optimisation : DB Indexes: Un-clustered Indexes

Indexed by some other Key



Comp283-Lecture 5
Optimisation: Table Partitioning

- Used for Large databases.
- Table Partitioning cuts up the table into smaller more manageable sizes.
- The DBMS performs Partitioning in the background.
- Improves performance for queries on small subsets of data.
- Hardware Partitioning
- Horizontal Partitioning
- Vertical Partitioning
  - Normalisation
  - Row Splitting

Comp283-Lecture 5

DB Storage: Normalisation Forms

- Normalisation is the process of taking all of the data that is to be stored in a database and separating it into distinct tables following strict rules to produce the most efficient database storage structure.
- Entity integrity
  - Each row of a table is consistent, generally it should represent a set of unique attributes (thus it is a Tuple) structured so that each attribute represents meaning for the object
- Domain integrity
  - Each column of a table is consistent - all data in the column conforms to the definition of the attribute. A constraint can be placed on the column to force domain integrity. E.g. you can allow or disallow Null values, define a range within which the data value must exist, the format of the data, as well as specifying the data-type.

Comp283-Lecture 5

DB Storage: Normalisation Forms

- Referential integrity
  - Maintain the data consistency between columns in a table and also between related tables in the database. Often enforced using a foreign key which associates the data attributes with data attributes in another table.
- User-defined integrity
  - Maintain data consistency by stipulating other specific rules. Usually enforced using stored procedures and triggers. e.g. you may require a log table to be updated with a reference to a data insert, recording the user that made the insert - Can be done using a trigger on the data table that causes some code to be executed to update the log table.

Comp283-Lecture 5
DB Storage: 1NF

- The First Normal Form (1NF)
  - The tables must not have any duplicate records, creating a primary key on the table will satisfy this rule.
  - No two columns in the table should describe the same attribute type. e.g. in a banking system; Current, Savings, Loan columns are all describing bank account types.
  - Entries in a column must be of the same data-type.
  - Each row must contain the same number of columns (each record/tuple has the same number and type of attributes – just different values)

Comp283-Lecture 5
DB Storage: 2NF

- The Second Normal Form (2NF)
  - Must be in 1NF
  - Any non-candidate key attribute must be dependant on the whole of the Primary Key.
  - e.g.
    - {Flight_No, Date, Safety_Check, Destination} Fails 2NF
    - But
    - {Flight_No, Date, Safety_Check}
      and
      {Flight_No, Destination} Passes 2NF

Comp283-Lecture 5
DB Storage: 2NF

Flight_No	Date	Safety_Check	Destination
BA32765	21/05/2017	Full-OK-03632	Belfast
BA32765	22/05/2017	Full-OK-03784	Belfast
LH93162	22/05/2017	Part-OK-73291	Berlin
BA32765	23/05/2017	Full-OK-26534	Belfast

Flight_No	Date	Safety_Check
BA32765	21/05/2017	Full-OK-03632
BA32765	22/05/2017	Full-OK-03784
LH93162	22/05/2017	Part-OK-73291
BA32765	23/05/2017	Full-OK-26534

Flight_No	Destination
BA32765	Belfast
LH93162	Berlin

Comp283-Lecture 5
DB Storage: 3NF

- The Third Normal Form (3NF)
  - Must be in 1NF & in 2NF.
  - The tables must only contain non-primary key attributes that are fully dependant on the Primary Key.
  - e.g.
    - {Flight_No, Date, Gate_No, Gate_Supervisor} Fails 3NF, succeeds 1NF and 2NF
    - {Flight_No, Date, Gate_No} and {Gate_No, Date, Gate_Supervisor} are in 3NF

Comp283-Lecture 5
DB Storage: 3NF

Flight_No	Date	Gate_No	Gate_Supervisor
BA32765	21/05/2017	12	Mr Jones
BA32765	22/05/2017	12	Ms Smith
LH93162	22/05/2017	8	Mr Jones
BA32765	23/05/2017	12	Ms Brown

Comp283-Lecture 5
DB Storage: 3NF

Flight_No	Date	Gate_No
BA32765	21/05/2017	12
BA32765	22/05/2017	12
LH93162	22/05/2017	8
BA32765	23/05/2017	12

Date	Gate_No	Gate_Supervisor
21/05/2017	12	Mr Jones
22/05/2017	12	Ms Smith
22/05/2017	8	Mr Jones
23/05/2017	12	Ms Brown

- Our Date attributes would actually be Date and Time of course - unless we only had one flight per day per gate!

Comp283-Lecture 5
DB Storage: 4NF

- The Fourth Normal Form (4NF)
  - Must be in 1NF and in 2NF and in 3NF.
  - The tables must contain only one independent multi-valued fact about the entity.
    - e.g.
      {Person, CarReg, PetName} Fails 4NF
      A person could own more than 1 car and could own more than 1 pet, but cars and pets are not dependent on each other.
  - How do we record this?

Comp283-Lecture 5
DB Storage: 4NF - continued

- Possible solutions for {Person, Cars, Pets}

Person	Cars	Pets
Fred	Lamborghini Gallardo	
Fred		Prince Rupert 1 (dog)
Fred		Mollie (cat)
Fred	Citroen 2CV	

Person	Cars	Pets
Fred	Lamborghini Gallardo	Prince Rupert 1 (dog)
Fred	Citroen 2CV	Prince Rupert 1 (dog)
Fred	Lamborghini Gallardo	Mollie (cat)
Fred	Citroen 2CV	Mollie (cat)

Person	Cars	Pets
Fred	Lamborghini Gallardo	Prince Rupert 1 (dog)
Fred	Citroen 2CV	Mollie (cat)

Or conform to 4NF:

{Person, Cars} and
{Person, Pets}

Comp283-Lecture 5
DB Storage: 5NF

- The Fifth Normal Form (5NF)
  - Must be in 1NF, in 2NF, 3NF and in 4NF.
  - A specialisation of 4NF.
  - In this case some external rule governs the relationship of the facts to the entity
    - e.g.
      {Salesman, Manufacturer, Products}
      Rule: A salesperson authorised to sell products of type A and also authorised to sell any product from Manufacturer B, must be able to sell any product of type A made by manufacturer B.

Comp283-Lecture 5
DB Storage: 5NF - Continued

Salesman	Manufacturer	Product
Fred	Phillips	TV
Jane	Phillips	Amplifiers
Jane	BOSE	Speakers
Joe	Sony	TV
Joe	Sony	Amplifiers

{Salesman, Manufacturer, Products}

Rule: A salesperson authorised to sell products of type A and also authorised to sell any product from Manufacturer B, must be able to sell any product of type A made by manufacturer B.

Salesman	Product
Fred	TV
Jane	Amplifiers
Jane	Speakers
Joe	TV
Joe	Amplifiers

Salesman	Manufacturer
Fred	Phillips
Jane	Phillips
Jane	BOSE
Joe	Sony

Manufacturer	Product
Phillips	TV
Phillips	Amplifiers
BOSE	Speakers
Sony	TV
Sony	Amplifiers

COMP283-Lecture 5

Conclusion

- Database Keys
- Normalisation