CM1603 - Database Systems

Week 06| Relational Model - II

Dileeka Alwis – Lecturer / Level Coordinator,

Department of Computing, IIT

ROBERT GORDON UNIVERSITY ABERDEEN
TEF Gold
INFORMATICS INSTITUTE OF TECHNOLOGY

Learning Outcomes

- Covers LO1 for Module - Describe and evaluate underlying theory and principles of relational database management systems (RDBMS).
- Covers LO2 for Module – Analyses and apply database design and modelling methods for a given business case study
- On completion of this lecture, students are expected to be able to:
  - Convert EER diagram in to relational schema.
  - Define Integrity rules in RDBMS

Lesson Outline

- Convert EERD into Relational Schema
  - Generalisations with {Mandatory, AND}
  - Generalisations with {Mandatory, OR}
  - Generalisations with {Optional, AND}
  - Generalisations with {Optional, OR}

- Integrity Constraints in RDBMS
  - Domain Integrity
  - Key Integrity
  - Entity Integrity
  - Referential Integrity

Convert EERD into Relational Schema

- EERD is converted into Relational Schema by mapping following constraints:

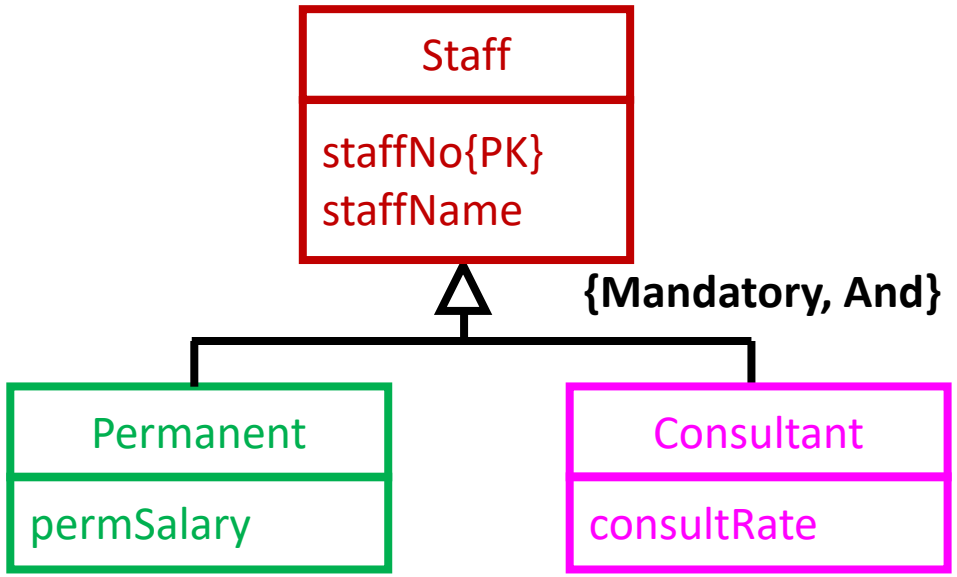
  - Generalisations with {Mandatory, AND}
  - Generalisations with {Mandatory, OR}
  - Generalisations with {Optional, AND}
  - Generalisations with {Optional, OR}

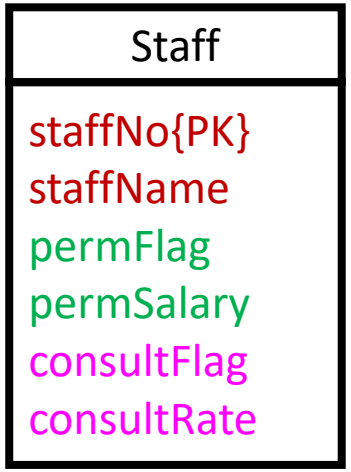
Generalisations with {Mandatory, AND}

- Merge all entities into **one table** with all attributes under new table.
- Create PK of the new table as the PK of the generalised entity.
- Add flags to differentiate between records of previous specialised entities.

Generalisations with {Mandatory, And}

Conceptual





Relational Schema Staff (staffNo{PK}, staffName, permSalary, consultRate, permFlag, consultFlag)

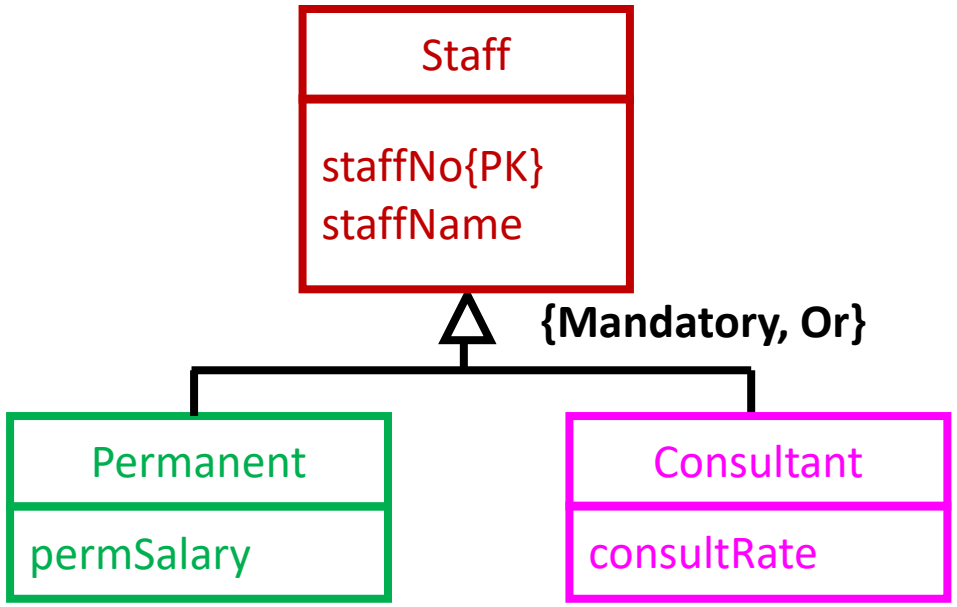
Generalisations with {Mandatory, OR}

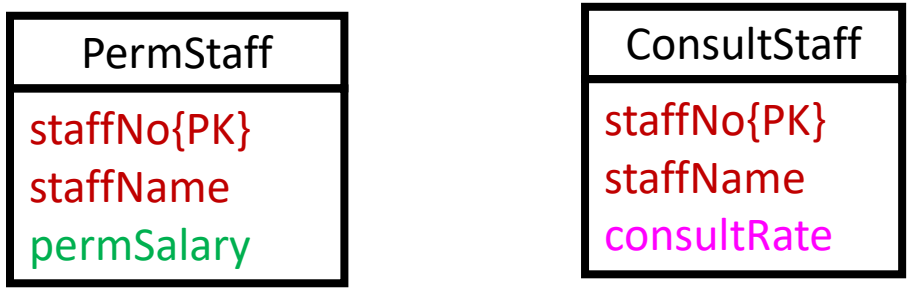
- Create **separate tables for each sub entity**
  - One table for each of the specialised entities with the attributes of the general entity also added.
  - PK for the tables is the PK of original general entity.

- Each table have their own relationships with the rest of the logical schema
  - The relationships that were associated to the original general entity are doubled up.
  - The relationships that were associated to each specialised entities remain the same.

Generalisations with {Mandatory, Or}

Conceptual





Relational Schema

PermStaff (staffNo{PK}, staffName, permSalary)
ConsutlStaff(staffNo{PK}, staffName, consultRate)

Generalisations with {Optional, AND}

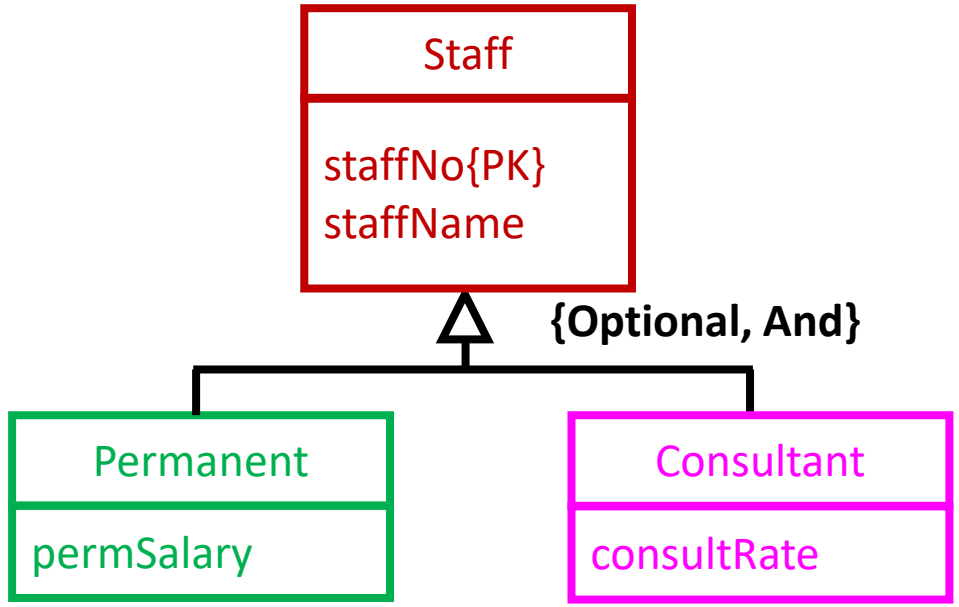
- Create **2 tables**
  - One table for general entity which becomes the Parent table.
  - One table for all the specialised entities merged together which becomes the Child table.

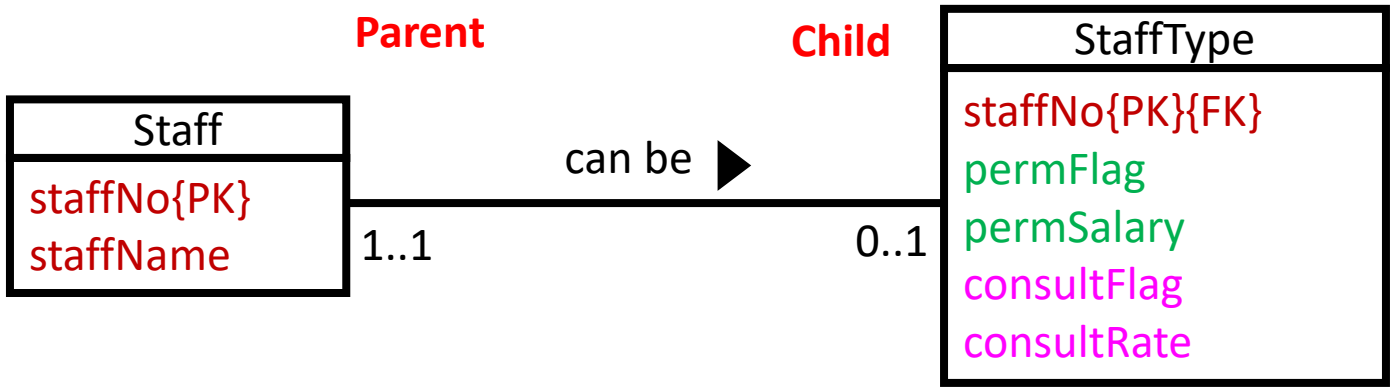
- Create ONE one-to-one relationship optional on one side between the 2 tables
  - FK of the Child table references PK of the Parent table.
  - PK of the Child table is the same as the PK of Parent table.

- Add flags to differentiate between records of previous specialised entities.

Generalisations with {Optional, And}

Conceptual





Relational Schema

Staff (staffNo{PK}, staffName)
StaffDetails (staffNo{PK}{FK}, permFlag, permSalary, consultFlag, consultRate)

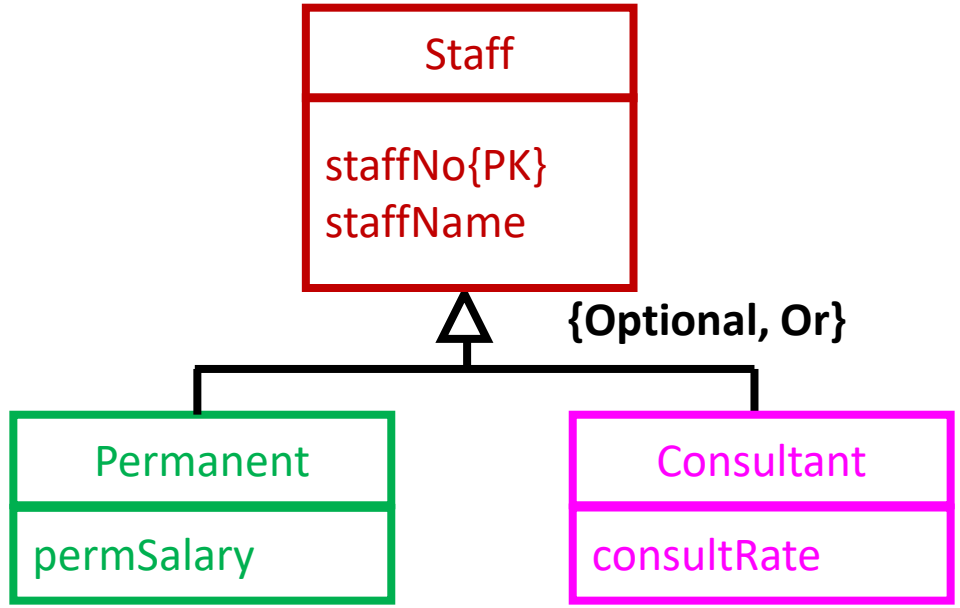
Generalisations with {Optional, OR}

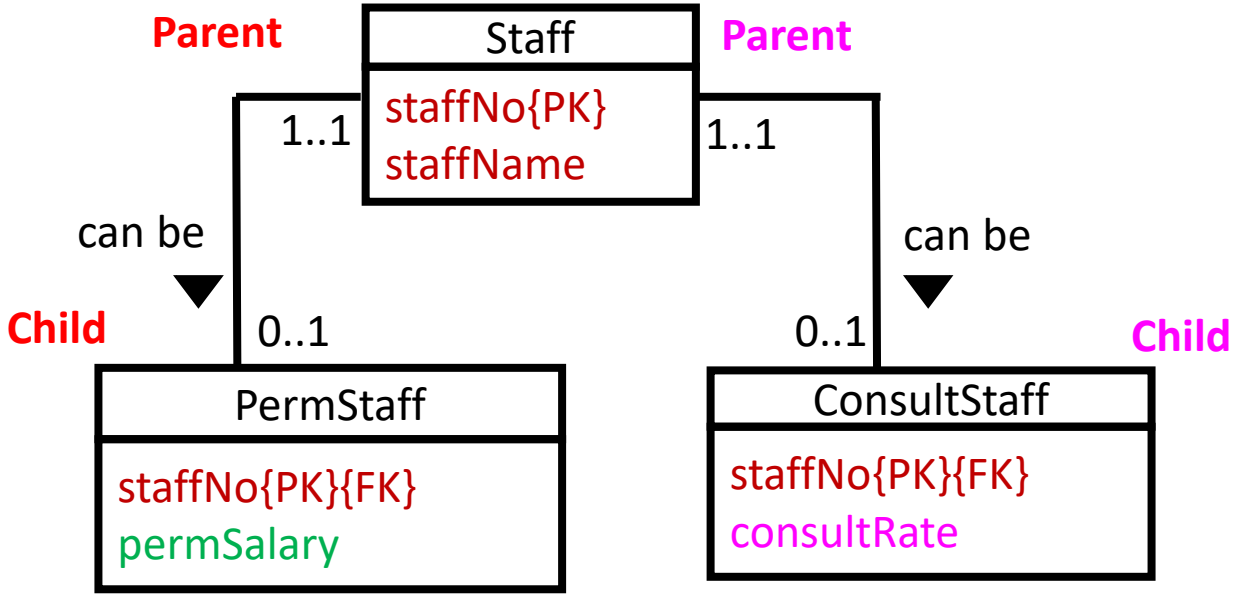
- Create **many tables**
  - One table for general entity which becomes the Parent table.
  - One table for each of the specialised entities, which becomes the Child table respectively.

- Create TWO one-to-one relationships optional on one side between the Parent table and the respective Child tables.
  - FK of each Child table references PK of the Parent table.
  - PK of each Child table is the same as the PK of Parent table.

Mapping Generalisations with {Optional, Or}

Conceptual





Step-by-step Approach to Relational Schema

1. Map specialisations
2. Map one-to-one relationships mandatory on both sides
3. Map n-ary & many-to-many relationships
4. Map one-to-many relationships and one-to-one relationships that are optional on one side or on both sides.

Integrity Constraints in RDBMS

- A **set of rules** or **constraints** that are **used to maintain the quality of data** stored in a Relational DB.
  - Limit or validate the type of data in a table.
  - Ensures the accuracy and reliability of the data.
- Can define validation rules or formats for the data to avoid values which are not acceptable (garbage values) for a particular table.
- Integrity rules are very important for a good database design.

Types of Integrity Rules

- Domain Integrity
- Key Integrity
- Entity Integrity
- Referential Integrity

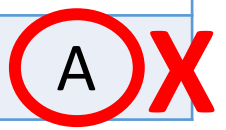
Domain Integrity

- Defines a valid set of values for an attribute.
  - Which values are allowed / not allowed.
  - Which is the format of values.

- A **domain** is a set values of the same data type.

Student

StudentID	Name	Age
1001	Tom	21
1002	John	20
1003	Ann	23
1004	John	20
1005	Mary	A



- The values of the attribute (domain) must be available in the corresponding data type and format.

Key Integrity

- Primary key **must be unique**.
- Primary key value is used to identify individual records in a relation uniquely.
- No two records can have the same combination of values for all their attributes.

Primary Key

StudentID	Name	Age
1001	Tom	21
1002	John	20
1002	Ann	23
1004	John	20
1005	Mary	22

X

Entity Integrity

- Every table **must** have a primary key.
- Either the primary key or any part of it **cannot contain null values**.
- **Null**
  - Attribute value is not relevant or unknown.
  - Does not mean that the value is blank or zero.
  - The system will fill it with an irrelevant value.

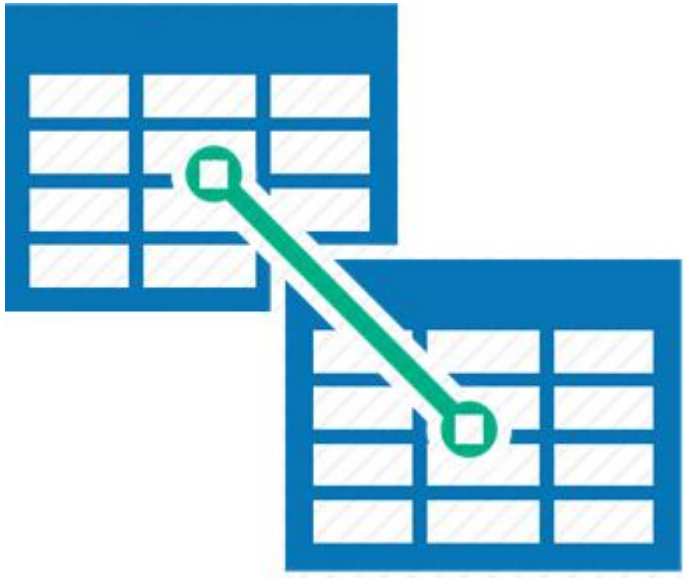
Primary Key

StudentID	Name	Age
1001	Tom	21
1002	John	20
1003	Ann	23
	John	20
1005	Mary	22

X

Referential Integrity

- Specified between two tables with **primary key** and **foreign key**.
- Data in the foreign key field of a particular table must have a matching primary key value in the other table, or the foreign key value can be null.
- It is important that both primary key and foreign key have the same data type and come from the same domain.



Example - Referential Integrity

Degree

DegreeCode	Degree Title
CS	Computer Science
SE	Software Engineering
CN	Computer Networks

Primary Key

Student

StudentID	Name	Age	Degree
1001	Tom	21	SE
1002	John	20	MIS
1003	Ann	23	SE
1004	John	20	Null
1005	Mary	A	CS

X

Foreign Key

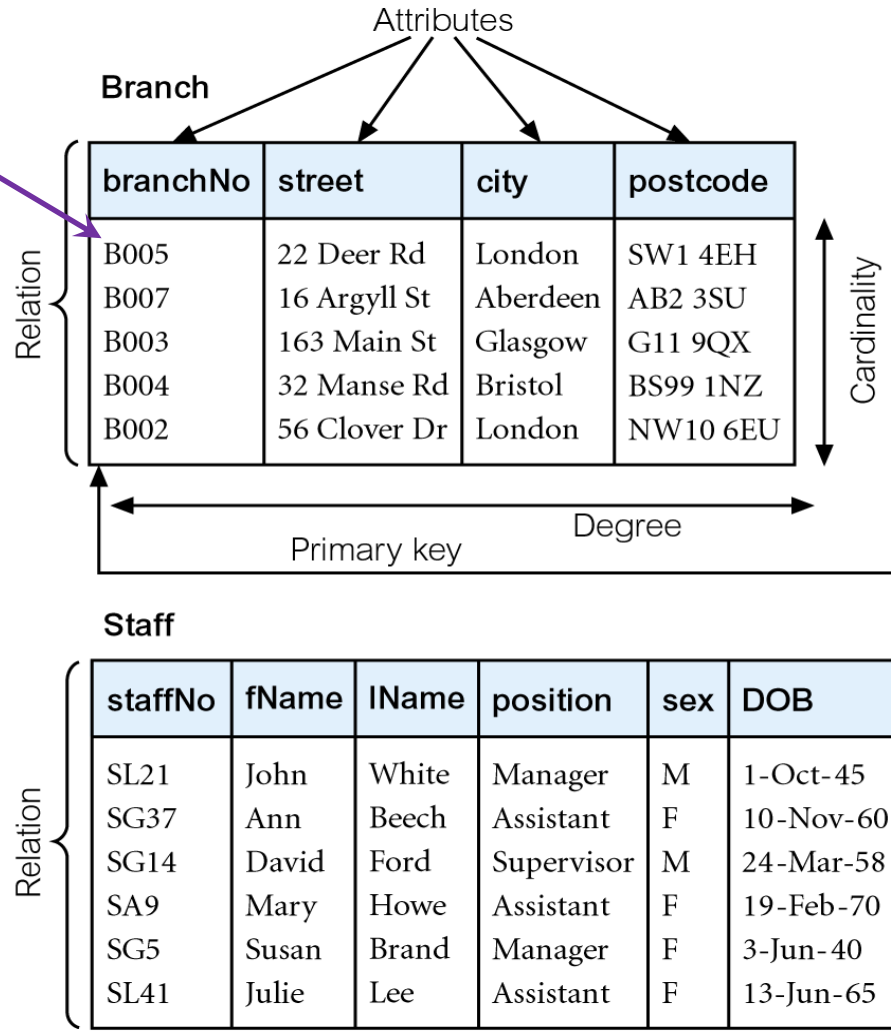
Primary Key

Enforce Referential Integrity

- It is possible for an attribute NOT to have a corresponding value, but it will be impossible to have an invalid entry.
- The enforcement of the referential integrity rule makes it impossible to update or delete a row in a table whose primary key has mandatory matching foreign key values in the another table.
- **DELETE CASCADE** - Automatically deletes the matching records in the child table when the corresponding parent record is deleted.
- **UPDATE CASCADE** - Automatically updates the matching records in the child table when the corresponding parent record is updated.

Relational Model – Recap

ENTITY INTEGRITY:
values of a PK can never be null!



REFERENTIAL INTEGRITY:
values of a FK can either be null or match values of the PK they reference, nothing else!

Thank you

Contact: dileeka.a@iit.ac.lk

ROBERT GORDON UNIVERSITY ABERDEEN
TEF Gold
INFORMATICS INSTITUTE OF TECHNOLOGY