Self-note:

* Partial Dependency occurs when a non-prime key is dependent on part of a prime key. E.g. teacher partially depends on COMPOSITE KEY, hence we have Course -> Teachers
* Transitive Dependency occurs when a non-prime key column value can be guessed or taken from other non-key column, i.e. MaxMark transitively depends on Exam Type (duplicated columns, identified by another student id column)

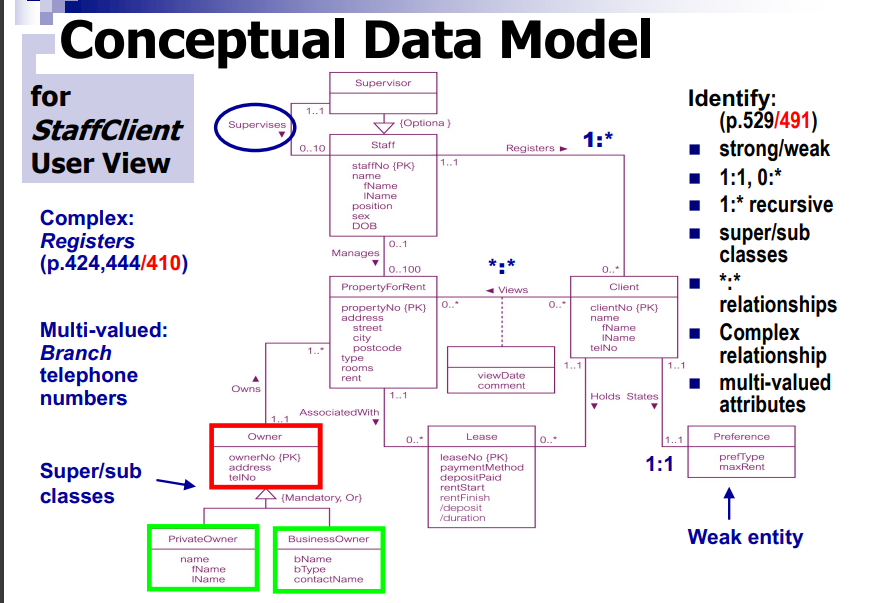
Semantic concepts are incorporated into the original ER model and called the Enahcned Entity-relationship (EER) model

Most useful additional concepts of EER model are called specialization/generalization, (aggregation and composition: not considered here).

* Superclass: entity type that includes on r more distinct subgrouping of its occurrences
* Subclass: A distinct subgrouping of occurrences of an entity type.

A diagram of a data flow

Description automatically generated



Check for derived attributes (are derived dynamically from other parts of the database)

In a table, there can only be one Primary key for a table. Therefore all the remaining Candidate keys are known as Alternate or Secondary keys

* A primary key cant be an alternate key. For a table with a single Candidate key which has to be the primary key will not contain any alternate key
* A foreign key cant be an alternate key as it is only used to reference another table
* The Alternate key should be unique
* An alternate key can be a set of a single attribute or multiple attributes
* It can be NULLs as well

Consider a tale customer information:

* Product id: primary key of table1
* CustomerId is the primary key of table 2
* Customer information is the combined table shown CustomerId, customerName, Email address, shipping address, pan number and product id
* Candidate key sets: Customer ID, Pan Number, Email Address

Say the database administrator of this E-Commerce site picked Customer ID as the primary Key. Therefore, PAN number and email address will be alternate keys or secondary keys. Alternate key has all the properties to become a primary key so is an alternate option.

1NF to 2NF:

* Identify the primary key for the 1NF relation
* Identify the functional dependencies in the relation
* If partial dependencies (of non-primary-key attributes) exists on the primary key, remove them by placing them in a new relation along with a copy of their determinant

2NF to 3NF:

* Aims to eliminate the transitive dependencies, as they can cause insertion and deletion and update anomalies.
* A relation is in 3NF if and only if, it is a relation in 2NF and in which no non-primary-key attribute is transitively dependent on the primary key.

**Instant Recuitment:**

Casual Staff:

* CasualStaffMember
* Home address
* Email address
* Mobile phone number
* Est.travelTime
* /
* SubjectCanTeach
* YearOfExperience
* Self-RateCompetitiveness/PerferenceWithTheSubject
* Qualifications
* Positions are interested in casual Work (Leturing, tutoring, or marking or etc).

Position:

* DifferentPositionAvailable (whether it is for lecturing, tutoring or marking). The casuals are paid on hourly basis according to their positions.

StaffRequest: When a casual staff is being requested or sought, it will be associated with a particular position for the payment rate, and the **position** will **specify certain expertise** or **subjects** the casual staff should be familiar with

* StaffRequestedID
* Position
* PayRate
* SubjectstheCasualStaffFamiliar

Recruitment: each casual staff is **recruited to a specific academic position**

* RecuimentPeriodOfTime
* CasualStaffID
* Venue

ExpressRecruitment:

* AvailableStaff
* RankBasedOnPastResponseTime #There is condition: expire after a pre-selected amount of response time
* Est.TravelTimetoSchool
* ContactRecommendations
* AcceptanceRecords (accepted/rejected)
* Time

Availability:

* StaffId
* AvailableWorkHours/Day/weeks

**Instant Recruitment** is a system to be designed to recruit casual staff under a very short notice, as well as on a more regular basis. For simplicity, this system will be limited to serving a large school at a university where casual staff need to be recruited for lecturing, tutoring, and marking etc

**Casual staff:** Within the school's recruitment database, each candidate for casual staff must provide their contact details in terms of home address, email address and mobile phone number, and must provide an estimated travel time to reach the school from home. Each candidate has to specify a list of subjects they can teach, their relevant years of experience there, and their self-rated competitiveness or preference with the subject in the range of 0 to 10. Each candidate also needs to enlist their qualifications and the positions they are interested for the casual work.

**Positions:** There can be different type of positions made available to the casuals, according to whether a position is for lecturing, tutoring or marking etc. The casuals are paid on hourly basis according to their positions.

**Staff Request:** When a casual staff is being requested or sought, it will be associated with a particular position for the payment rate, and the position will specify certain expertise or subjects the casual staff should be familiar with.

**Recruitment:** Each casual staff is recruited to a specific academic position for a period of time to conduct the teaching activity at a given venue, and will be assigned to a relevant academic supervisor for the academic liaison. For example, a casual staff may be employed to do 4 hours each week for the in-class tutoring during the semester, and this type of recruitment is considered a *regular* recruitment. For a **regular recruitment**, a casual staff will be assigned a specific class venue and the number of hours to work there starting from an allocated beginning time. The recruitment will also specify the number of weeks the appointed casual is to conduct the same activity at the same weekly time and venue.

*We note that for this****regular part****of Assignment 2, students do not need to read the rest of the case description. Moreover the casual staff can be assumed to be available all the time other than those time slots already booked by this same recruitment system for another casual work. The job offer/acceptance tracking, or the****negotiation****between the School and the casuals, doesn't have to be considered either.*

We note that the **estimated travel time** may be further utilized or taken into consideration when the *Instant Recruitment System* suggests who to send first the request to stand in for a regular staff who can't make it due to for instance a sudden sickness.

**Express Recruitment:** A casual staff may, for instance, also be urgently recruited to conduct a 2 hour lecture at a certain time on the day under a very short notice (hence an *express* recruitment).

For an **express recruitment**, the system will first search the available casual staff and rank them according to their past response time and their estimated travel time to reach the school. A school's administrative staff will usually select one of the top recommendations to contact via phone or SMS or email. A staff request (i.e. a job offer) sent via an email or SMS to a casual staff will expire after a pre-selected amount of response time associated with this particular job, so that the admin can select the next candidate to contact for the casual position. For any express recruitment, the relevant communications between the school and the casual staff will be recorded and will be later analysed to calculate his or her average response time. In particular, the recruitment system needs to be able to track when a casual job is sent to whom, whether it is accepted or rejected, and at what time etc.

**Availability:** Each casual staff may be available only on certain days of the week, and may also be unavailable for certain specific periods of days.

# Question 1: Selected Additional Exercise

## 1/Complete Question alpha and gamma in the additional Exercises for practical 8

### Question alpha:

Data anomaly occurs when alternation of records in the redundant data are not made successfully. Deletion anomalies occur as when deleting records from the redundant data results in the accidental loss of data.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **StoreID** | **StoreName** | **StorePostcode** | **SalePerson** | **Number** |
| 001 | Aukland | 2145 | Mary Hatson | 043-222-435 |
| 001 | Aukland | 2145 | John Dean | 043-261-413 |
| 003 | Pokart | 2133 | Garp Fisherman | 042-260-222 |
| 004 | Aokmond | 2000 | Skylar White | 041-250-222 |
| 004 | Aokmond | 2000 | Walter White | 041-250-221 |
| 002 | SpringField | 2010 | Felina | 041-250-099 |

For instance: In the 3rd row, deleting Garp Fisherman and his number out of the store as he is planning to retire will result in losing information about the existence of Pokart store, including its store id, store name and as well as the store post code.

### Question gamma:

## 2/Complete Question theta in the additional exercises for Practical 9

### Question theta:

## 3/Complete Question alpha and Beta in the additional Exercises for Practical 10

### Question alpha:

Tables:

|  |  |  |  |
| --- | --- | --- | --- |
| **TrainID** | **TrainNumber** | **BranchNo** | **BranchOperator** |
| 1 | 1421 | T01 | John Dean |
| 2 | 2222 | T02 | Marie |
| 3 | 2468 | T02 | Marie |
| 4 | 1357 | T03 | Rita |

|  |  |  |  |
| --- | --- | --- | --- |
| **CustomerID** | **CustomerName** | **PhoneNumber** | **Postcode** |
| 1 | Quang Dong Nguyen | 030-246-137 | 2150 |
| 2 | Rakha Adam | 030-245-136 | 2150 |
| 3 | Aws Alzuhairi | 030-244-135 | 2140 |
| 4 | Mousa Raza | 030-241-133 | 2134 |

**Tables with Functional Dependency, attributes and values:**

A diagram of a number and number

Description automatically generated with medium confidence

A diagram of a number of numbers

Description automatically generated

**Determine Functional Dependency in each of the table:**

2NF:

**TrainID** -> TrainNumber, BranchNo, BranchOperator

**CustomerID** -> CustomerName, PhoneNumber, Address

3NF:

**TrainID** -> TrainNumber, BranchNo

**BranchNo** -> BranchOperator

**CustomerID** -> CustomerName, PhoneNumber, Address

## 4/ Complete Questions alpha and Beta in the additional Exercises for Practical 11

### Question alpha:

In the context of database transactions, all transactions must display Atomicity, Consistency, Isolation and Durability (often abbreviated to ACID). In short, these are the fundamental steps toward of a successful transaction. Each of these properties means:

* Atomicity: All operations of transactions must be completed as a whole, or none will be executed. Atomicity helps to protect the integrity of the database and to prevent corruption in transactions.
* Consistency: Ensures the outcomes are yielded with consistent results from the concurrent execution of multiple transactions. Consistent results refer to the absolute adaptability of results to all the business rules, database rules and constraints. And if it fails to converge the adaptation, meaning the data drive into an illegal state during the transaction, then the whole transaction will be discarded.
* Isolation: Ensures all transactions execute independently of one another in an isolated environment, meaning that transactions can be executed simultaneously, and no data should have an impact on another. This isolated environment enables both transactions guaranteed to reach completion while not interfering with each other.
* Durability: Indicates the permanent state of data’s consistency. In another word, when a transaction is completed, the database reaches a consistent state, meaning changes in data are permanently recorded in the database and it will not be lost because of any events of disruption such as system’s failure, power outages, etc.

An example to demonstrate the property of the letter “I” in ACID:

A banking application where two concurrent transactions occur at the same time. Transaction 1 has a monetary transfer of 500$ from account 1 to account 2, while the second transaction shows a withdraw of 100$ from account 1. Based on the Isolation properties of database, it is guaranteed to succeed both transactions. However, it will go as follows: The first transaction is proceeded if there is no systematic errors or logical error happened in the database system, then it will be completed successfully and followed on by the second transaction. Since the data required is based on the first transaction, there will be no second stage of transaction if the first transaction meets an error. In other word, the process on the second state will begin after the operation on the first state of transaction is finished. Therefore, if the second transaction is completed without errors, then the total operation succeeds; Otherwise, the whole transaction operation fails to complete and there will be no transaction is made.

### Question beta:

# Question 2:

A computer screen shot of a diagram

Description automatically generated

A computer screen shot of a computer program

Description automatically generated