Group 10 LD-01

Ariel Setiawan – 2201794574

Chris Andre Sitompul – 2201798944

Michael Yehezkiel Lie – 220180587

Tiffani Francisca Kizana – 2201728023

**Step 1.1. Identify entity types**

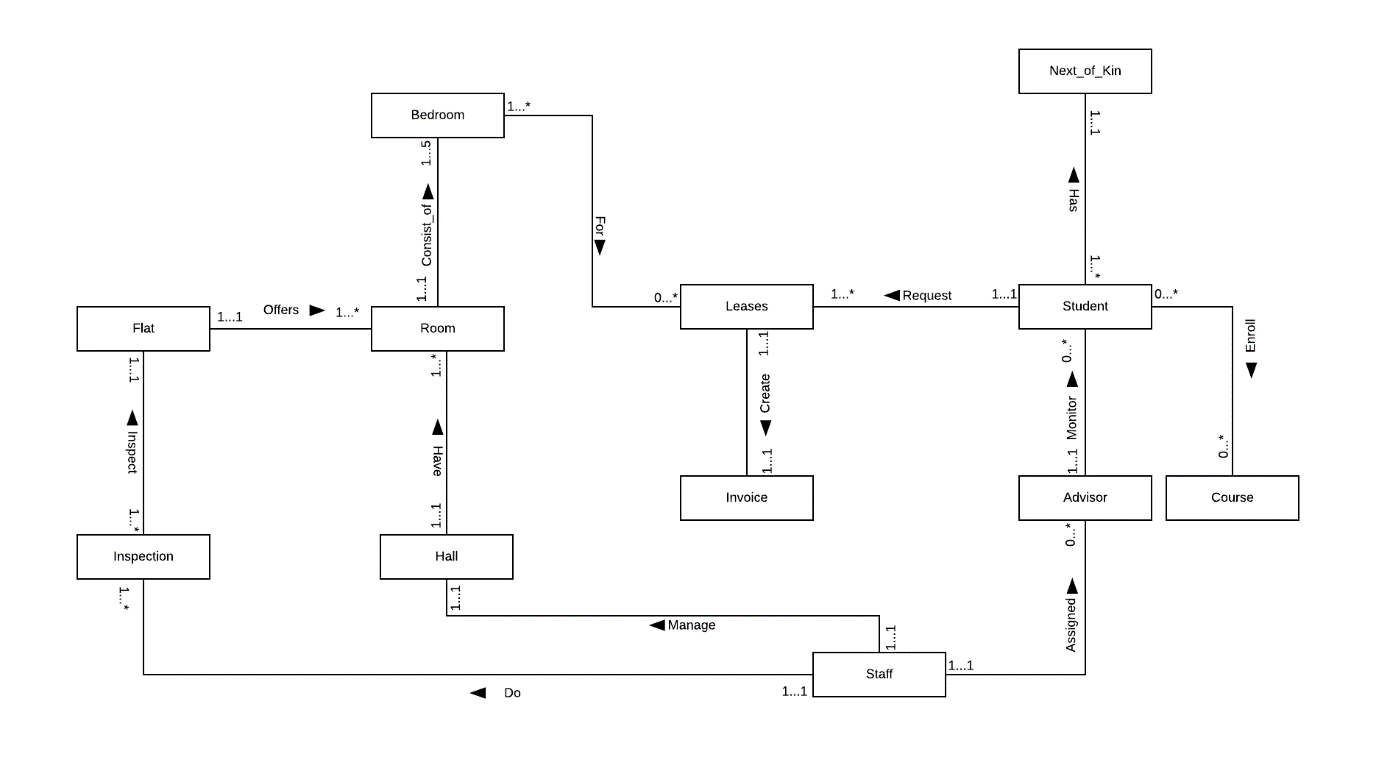
* Students
* Advisor
* Hall
* Flat
* Bedroom
* Leases
* Invoice
* Inspection
* Staff
* Course
* Next\_of\_Kin
* Room

**Step 1.2. Identify relationship types**

|  |  |  |  |
| --- | --- | --- | --- |
| Entity Name | Description | Aliases | Occurrence |
| Students | Istilah umum untuk seluruh mahasiswa *full-time* dalam Universitas | Student | Setiap mahasiswa dapat menyewa ruangan di Hall of Residence atau di Apartment.  Setiap mahasiswa bisa terdaftar di daftar tunggu sewa ruangan*.*  Setiap mahasiswa diawasi oleh seorang *advisor*.  Setiap Mahasiswa bisa mendaftar matakuliah.  Setiap mahasiswa bisa mendaftarkan kerabatnya. |
| Advisor | Istilah umum yang digunakan untuk mendeskripsikan seluruh staff yang bertindak sebagai *advisor* mahasiswa yang terdapat dalam universitas | Advisor | Setiap mahasiswa memiliki satu *advisor* |
| Hall | Istilah umum untuk mendeskripsikan *hall* yang dapat dijadikan tempat tinggal di Universitas | Hall | Setiap *hall* memiliki banyak ruangan.  Setiap ruangan dapat disewa oleh seorang mahasiswa.  Setiap *hall* di kelola oleh seorang manager *Hall.*  *Setiap* ruanganmemiliki 1 *bedrooms.* |
| Flat | Istilah umum untuk mendeskripsikan *apartment* yang dapat dijadikan tempat tinggal oleh mahasiswa | Flat | Setiap *flat* terdiri dari banyak *single-room*.  Setiap single-room terdiri dari 3-5 *bedrooms.*  Setiap *bedroom* di sewa oleh satu mahasiswa. |
| Bedroom | Isilah umum untuk mendeskripsikan kamar tidur yang terdapat pada ruangan di flat atau hall | Bedroom | Setiap ruangan di hall memiliki 1 bedrooms.  Setiap ruangan di flat memiliki 3 – 5 bedrooms. |
| Leases | Istilah umum untuk menggambarkan seluruh data persetujuan sewa ruangan di Hall atau Flat | Lease | Setiap mahasiswa dapat menyewa ruangan dengan minimal sewa 1 semester, dan maksimal sewa 1 tahun. |
| Invoice | Istilah umum untuk mendeskripsikan faktur pembayaran antara mahasiswa dan *resident office* | Invoice | Setiap mahasiswa dikirim sebuah *invoice* saat awal semester. |
| Inspection | Istilah umum untuk mendeskripsikan seluruh proses inspeksi yang dilakukan oleh staff | Inspection | Setiap inspeksi dilakukan oleh seorang staff. |
| Staff | Istilah umum untuk mendeskripsikan seluruh karyawan yang bekerja pada *residence office* | Staff | Setiap hall manager mengawasi operasi di *hall.* Setiap staff dapat melakukan banyak inspeksi.  Ada staff yang di tugaskan sebagai advisor mahasiswa |
| Courses | Istilah umum yang digunakan untuk mendeskripsikan seluruh mata kuliah yang terdapat dalam universitas | Course | Setiap *course* dapat dipilih/dihadiri oleh banyak mahasiswa. |
| Next\_Of\_Kin | Istilah umum untuk mendeskripsikan kerabat yang dimiliki oleh mahasiswa | Next\_of\_Kin | Setiap mahasiswa dapat memiliki banyak kerabat. |
| Room | Istilah umum yang digunakan untuk mendeskripsikan ruangan tempat tinggal yang dapat disewa oleh mahasiswa | Room | Setiap ruangan dapat disewa oleh seorang mahasiswa  Setiap ruangan di hall memiliki 1 bedrooms.  Setiap ruangan di flat memiliki 3 – 5 bedrooms.  Setiap Hall memiliki banyak ruangan.  Setiap flat memiliki banyak ruangan. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity Name | Multiplicity | Relationship | Multiplicity | Entity Name |
| Student | 0..\* | Enroll | 0..\* | Course |
|  | 1..\* | Has | 1..1 | Next\_of\_Kin |
|  | 1..1 | Request | 1..\* | Lease |
| Advisor | 1..1 | Monitor | 0..\* | Student |
| Staff | 1..1 | Assigned | 0..\* | Advisor |
|  | 1..1 | Manage | 1..1 | Hall |
|  | 1..1 | Do | 1..\* | Inspection |
| Lease | 1..1 | Generate | 1..1 | Invoice |
| Bedroom | 1..\* | For | 0..\* | Leases |
| Hall | 1..1 | Have | 1..\* | Room |
| Inspecton | 1..\* | Inspect | 1..1 | Flat |
| Flat | 1..1 | Offers | 1..\* | Room |
| Room | 1..1 | Consist\_of | 1..5 | Bedroom |

**First Cut**



**Step 1.3. Identify and associate attribute with entity or relationship type**

Student :

1. Nomor siswa
2. Nama (composite : nama depan dan belakang)
3. Alamat rumah ( composite : jalan, kota, kode pos)
4. Nomor ponsel
5. E-mail
6. Tanggal lahir
7. Jenis kelamin
8. Kategori siswa
9. Kebangsaan
10. Kebutuhan khusus
11. Komentar tambahan
12. Status
13. Jurusan
14. Spesifikasi Jurusan

Advisor :

1. Nama departemen
2. Nomor telepon
3. E-mail

Hall:

1. Nomor hall
2. Nama Hall
3. Alamat
4. NoTelp

Room :

1. Nomor kamar

Bedroom :

* 1. Nomor tempat
  2. Biaya Sewa

Flat :

1. Nomor apartemen
2. Alamat

Leases :

1. Nomor sewa
2. Durasi sewa
3. Tanggal yang diinginkan siswa untuk masuk ruangan
4. Tanggal yang diinginkan siswa untuk meninggalkan ruangan (jika diketahui)

Invoices :

1. Nomor faktur
2. Semester
3. Jatuh tempo pembayaran
4. Metode pembayaran (cek, uang tunai, Visa, dan sebagainya)
5. Tanggal pengingat pertama dan kedua dikirim [1..2]

Inspection:

1. Tanggal inspeksi
2. Status property
3. Comment

Staff :

1. Nomor staf
2. Nama (composite : nama depan dan belakang)
3. E-mail
4. Alamat rumah (composite : jalan, kota, kode pos)
5. Tanggal lahir
6. Jenis kelamin
7. Posisi (misalkan : Manajer Hall, Asisten Administrasi, Pembersih)
8. Lokasi ( misalkan :Kantor atau Hall Residence)

Courses :

1. Nomor program
2. Nama program (termasuk tahun)
3. Instruktur kursus
4. Nomor telepon instruktur di kampus
5. E-mail instruktur
6. Nama departemen

NOK :

1. Nama keluarga dekat
2. Hubungan Antara keluarga dekat
3. Alamat (composite : jalan, kota, kode pos) keluarga dekat
4. Nomor telepon kontak anggota keluarga

**Step 1.4. Determine attribute domains**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entity Name | Attributes | Description | Data Type & Length | Nulls | Multi-Valued |
| Student | Banner\_No  Name  FName  LName  HomeAddress  Street  City  Postcode  PhoneNumber  Email  DOB  Gender  Category  Nationality  Special\_Needs  Comment  AccStatus  Major  Minor | Nomor Unik yang mengidentifikasi Mahasiswa  Nama pertama mahasiswa  Nama terakhir mahasiswa  Nama jalan alamat mahasiswa  Kota alamat mahasiswa  Nomor Kotak Pos alamat mahasiswa  Nomor mobile phone mahasiswa  Alamat email mahasiswa  Tanggal lahir mahasiswa  Jenis kelamin mahasiswa  Category mahasiswa( first-year, undergraduate)  Kebangsaan mahasiswa  Informasi kebutuhan khusus mahasiswa  Informasi tambahan mengenai mahasiswa  Status penempatan ruang kamar (placed/waiting)  Informasi major yang diambil oleh mahasiswa  Informasi minor yang diambil oleh mahasiswa | 10 character  100 character  100 character  255 character  100 character  10 character  20 character  100 character  Date  1 character (M atau F)  50 character  50 character  100 character  255 character  10 character  100 character  100 character | No  No  No  No  No  No  No  No  No  No  No  No  Yes  Yes  No  No  No | No  No  No  No  No  No  No  No  No  No  No  No  No  No  No  No  No |
| Advisor | Department  Int\_TelpNo  Email | Bagian tempat advisor bekerja  Telepon internal yang dimiliki advisor  Email yang dimiliki oleh advisor | 100 character  20 character  100 character | No  No  No | No  No  No |
| Hall | Hall\_No  Hall\_Name  Telp\_No | Nomor unik untuk setiap hall yang ada  Nama hall  Telepon setiap hall | 5 character  50 character  20 character | No  No  No | No  No  No |
| Flat | Flat\_No | Nomor unik setiap apartmen | 5 character | No | No |
| Bedroom | Place\_No  Rent | Nomor unik kamar tidur  Biaya sewa kamar tidur | 5 character  smallmoney | No  No | No  No |
| Leases | Lease\_No  Lease\_duration  Start\_date  Leave\_date | Nomor unik untuk surat keterangan sewa  Durasi sewa per semester  Tanggal mulai sewa  Tanggal selesai sewa | 5 character  integer  Date  Date | No  No  No  No | No  No  No  No |
| Invoice | Invoice\_No  Semester  Payment\_duedate  Payment\_date  Payment\_Method  Reminder[1..2] | Nomor unik untuk setiap bukti pembayaran  Semester pembayaran dilakukan  Batas akhir pembayaran  Tanggal pembayaran dilakukan  Metode pembayaran yang digunakan  Pengingat untuk batas akhir pembayaran | 5 character  Integer  Date  Date  20 character  date | No  No  No  No  No  Yes | No  No  No  No  No  Yes |
| Inspection | Inspection\_date  Property\_status  Comments | Tanggal pelaksanaan inspeksi  Status dari property yang diinspeksi apakah satisfying  Komentar tambahan hasil inspeksi | Date  10 character  255 character | No  No  Yes | No  No  No |
| Staff | Staff\_no  Name  FName  LName  Email  Home\_Address  Street  City  PostCode  DOB  Gender  Position  Location | Nomor unik untuk mengidentifikasi staff  Nama depan staff  Nama belakang staff  Email staff  Nama jalan alamat staff  Nama kota alamat staff  Kode pos alamat staff  Tempat tanggal lahir staff  Jenis kelamin staff  Posisi staff (manager, administrative assistant, cleaner)  Lokasi staff bekerja (residence hall, university) | 5 character  100 character  100 character  100 character  255 character  100 character  10 character  Date  1 character (M atau F)  50 character  50 character | No  No  No  No  No  No  Yes  No  No  No  No | No  No  No  No  No  No  No  No  No  No  No |
| Course | Course\_No  Course\_Name  Course\_Instructor  Int\_telpNo  Instructor\_Email  DepartmentName | Nomor unik untuk mata kuliah  Nama mata kuliah  Nama pengajar mata kuliah  Nomor telepon internal pengajar  Email pengajar  Nama departemen mata kuliah | 5 character  50 character  100 character  20 character  100 character  50 character | No  No  No  No  No  No | No  No  No  No  No  No |
| Next\_of\_Kin | Name  Relationship  Address  Street  City  Postcode  Telp\_No | Nama kerabat / keluarga  Hubungan kerabat  Nama jalan alamat kerabat  Nama kota alamat kerabat  Kode pos alamat kerabat  Nomor telepon kerabat | 100 character  50 character  255 character  100 character  10 character  100 character | No  No  No  No  Yes  No | No  No  No  No  No  No |
| Room | Room\_No | Nomor unik untuk ruangan | 5 character | No | No |

**Step 1.5. Determine candidate, primary, and alternate key attribute**

Attribut key yang terdapat dalam setiap entitas :

Student :

* Candidate Key : Banner\_No, FName, E-mail
* Primary Key : Banner\_No
* Alternative Key : FName, E-mail

Advisor :

* Candidate Key : Staff\_No, Int\_TelpNo, Email
* Primary Key : Staff\_No
* Alternative Key : Int\_TelpNo, Email

Hall :

* Candidate Key : Hall\_No, Telp\_No
* Primary Key : Hall\_No
* Alternative Key : Telp\_No

Room:

* Candidate Key : RoomNo
* Primary Key : RoomNo
* Alternative Key : -

Flat:

* Candidate Key : Flat\_No
* Primary Key : Flat\_No
* Alternative Key : -

Bedroom:

* Candidate Key : Place\_No
* Primary Key : Place\_No
* Alternative Key : -

Leases :

* Candidate Key : Lease\_No
* Primary Key : Lease\_No
* Alternative Key : -

Invoice :

* Candidate Key : Invoice\_No
* Primary Key : Invoice\_No
* Alternative Key : -

Inspection:

* Candidate Key : Inspection\_Date
* Primary Key : Inspection\_Date
* Alternative Key : -

Staff:

* Candidate Key : StaffNo, Email
* Primary Key : StaffNo
* Alternative Key : Email

Courses :

* Candidate Key : Course\_No
* Primary Key : Course\_No
* Alternative Key : -

Next\_of\_kin :

* Candidate Key : Name
* Primary Key : Name
* Alternative Key : -

**Step 1.6. Consider use of enhanced modeling concepts (optional step)**

(We skipped this step because there is no more enhance needed)

**Step 1.7. Check model for redundancy**

Leases and Invoices table is possible to be more specified to avoid anomaly & redundancy. Leases table still have attribute that can be moved to invoices to reduce redundancy and we can implement header and detail table to avoid anomaly between Leases and Invoices table.

**Step 1.8. Validate conceptual data model against user transactions**

1. Present a report listing the Manager’s name and telephone number for each hall of residence.
2. Present a report listing the names and banner numbers of students with the details of their lease agreements.
3. Display the details of lease agreements that include the summer semester.
4. Display the details of the total rent paid by a given student.
5. Present a report on students who have not paid their invoices by a given date.
6. Display the details of apartment inspections where the property was found to be in an unsatisfactory condition.
7. Present a report of the names and banner numbers of students with their room number and place number in a particular hall of residence.
8. Present a report listing the details of all students currently on the waiting list for accommodation; that is; who were not placed.
9. Display the total number of students in each student category.
10. Present a report of the names and banner numbers for all students who have not supplied details of their next-of-kin.
11. Display the name and internal telephone number of the Advisor for a particular student.
12. Display the minimum, maximum, and average monthly rent for rooms in residence halls.
13. Display the total number of places in each residence hall.
14. Display the staff number, name, age, and current location of all members of the residence staff who are over 60 years old today

A close up of a map

Description automatically generated

**Step 2 Build Logical Data Model**

**Step 2.1 Derive relation for logical data model**

1. **Strong entity types**
2. Staff (Staff\_No, FName, LName, Email, Street, City, Postcode, DOB, Gender, Position, Location)

Primary Key : Staff\_No

Alternate Key : Email

1. Flat (Flat\_No, Address)

Primary Key : Flat\_No

1. Hall (Hall\_No, Hall\_Name, Telp\_No, Address)

Primary Key : Hall\_No

Alternate Key : Telp\_No

1. Student (Banner\_No, FName, LName, Street, City, Postcode, PhoneNumber, Email, DOB, Gender, Category, Nationality, Special\_Needs, Comment, AccStatus, Major, Minor)

Primary Key : Banner\_No

Alternate Key : PhoneNumber, Email

1. Course (Course\_No, Course\_Name, Course\_Instructor, instructorEmail, instructorPhone, roomNo, departmentNo)

Primary Key : Course\_No

Alternate Key : Course\_Instructor, instructorEmail, instructorPhone

1. **Weak entity types**
2. Inspection (Inspection\_Date, Property\_Status, Comments)

Primary Key : none

1. Room (Room\_No)

Primary Key : Room\_No

1. Bedroom (Place\_No, Rent)

Primary Key : Place\_No

1. Leases (Lease\_No, Lease\_Duration, Start\_Date, Leave\_Date)

Primary Key : Lease\_No

1. Invoice (Invoice\_No, Semester, Payment\_duedate, Payment\_date, Payment\_Method, Reminder)

Primary Key : Invoice\_No

1. NOK (NOK\_Name, Relationship, Street, City, Postcode, Telp\_No)

Primary Key : NOK\_name

Alternate Key : Telp\_No

1. Advisor (Department, Int\_TelpNo, Email)

Primary key : none

1. **One-to-many (1:\*) binary relationship types**
2. Post Flat\_No from Flat into Inspection

Flat (Flat\_No, Address)

Primary Key : Flat\_No

Inspection (Inspection\_Date, Property\_Status, Comments, Flat\_No)

Primary Key : Flat\_No

Foreign Key : Flat\_No references Flat

1. Post Flat\_No from Flat into Room

Flat (Flat\_No, Address)

Primary Key : Flat\_No

Room (Room\_No, Flat\_No)

Primary Key : Room\_No

Foreign Key : Flat\_No references Flat

1. Post Staff\_No from Staff into Inspection

Staff (Staff\_No, FName, LName, Email, Street, City, Postcode, DOB, Gender, Position, Location)

Primary Key : Staff\_No

Alternate Key : Email

Inspection (Inspection\_Date, Property\_Status, Comments, Flat\_No, Staff\_No)

Primary Key : Flat\_No

Foreign Key : Flat\_No references Flat

Staff\_No references Staff

1. Post Hall\_No from Hall into Room

Hall (Hall\_No, Hall\_Name, Telp\_No, Address)

Primary Key : Hall\_No

Alternate Key : Telp\_No

Room (Room\_No, Flat\_No, Hall\_No)

Primary Key : Room\_No

Foreign Key : Flat\_No references Flat

Hall\_No references Hall

1. Post Staff\_No from Staff into Advisor

Staff (Staff\_No, FName, LName, Email, Street, City, Postcode, DOB, Gender, Position, Location)

Primary Key : Staff\_No

Alternate Key : Email

Advisor (Department, Int\_TelpNo, Email, Staff\_No)

Primary Key : Staff\_No

Foreign Key : Staff\_No references Staff

1. Post Staff\_No from Advisor into Student

Advisor (Department, Int\_TelpNo, Email, Staff\_No)

Primary Key : Staff\_No

Foreign Key : Staff\_No references Staff

Student (Banner\_No, FName, LName, Street, City, Postcode, PhoneNumber, Email, DOB, Gender, Category, Nationality, Special\_Needs, Comment, AccStatus, Major, Minor, StaffNo)

Primary Key : Banner\_No

Alternate Key : PhoneNumber, Email

Foreign Key : Staff\_No references Advisor

1. Post NOK\_Name from NOK into Student

NOK (NOK\_Name, Relationship, Street, City, Postcode, Telp\_No)

Primary Key : NOK\_name

Alternate Key : Telp\_No

Student (Banner\_No, FName, LName, Street, City, Postcode, PhoneNumber, Email, DOB, Gender, Category, Nationality, Special\_Needs, Comment, AccStatus, Major, Minor, StaffNo, NOK\_Name)

Primary Key : Banner\_No

Alternate Key : PhoneNumber, Email

Foreign Key : Staff\_No references Advisor

NOK\_Name references NOK

1. Post Banner\_No from Student into Leases

Student (Banner\_No, FName, LName, Street, City, Postcode, PhoneNumber, Email, DOB, Gender, Category, Nationality, Special\_Needs, Comment, AccStatus, Major, Minor, StaffNo, NOK\_Name)

Primary Key : Banner\_No

Alternate Key : PhoneNumber, Email

Foreign Key : Staff\_No references Advisor

NOK\_Name references NOK

Leases (Lease\_No, Lease\_Duration, Start\_Date, Leave\_Date)

Primary Key : Lease\_No

Foreign Key : Banner\_No references Student

1. **One-to-one (1:1) binary relationship types**
2. **Mandatory participation on both sides of 1:1 relationship**

* Leases dan Invoice

Sebelum di merge :

Leases (Lease\_No, Lease\_Duration, Start\_Date, Leave\_Date)

Primary Key : Lease\_No

Foreign Key : Banner\_No references Student

Invoice (Invoice\_No, Semester, Payment\_duedate, Payment\_date, Payment\_Method, Reminder)

Primary Key : Invoice\_No

Sesudah di merge :

Leases (Lease\_No, Lease\_Duration, Start\_Date, Leave\_Date, Invoice\_No, Semester, Payment\_duedate, Payment\_date, Payment\_Method, Reminder)

Primary Key : Lease\_No

Alternate Key : Invoice\_No

Foreign Key : Banner\_No references Student

1. **Mandatory participation on one side of a 1:1 relationship**

* Staff (child) dan Hall (parent)

**Ralat :** hendaknya multiplicity yang dimiliki oleh staff dan hall yakni

Hall 0..1 --- 1..1 Staff

Karena tidak setiap staff harus memanage sebuah hall

Hall (Hall\_No, Hall\_Name, Telp\_No, Address)

Primary Key : Hall\_No

Alternate Key : Telp\_No

Staff (Staff\_No, FName, LName, Email, Street, City, Postcode, DOB, Gender, Position, Location, Hall\_No)

Primary Key : Staff\_No

Alternate Key : Email

Foreign Key : Hall\_No references Hall

1. **Optional participation on both sides of a 1:1 relationship**

--- Tidak ada ---

1. **One-to-one (1:1) recursive relationship**

--- Skip karena tidak ada relasi yang rekursif --

1. **Superclass/subclass relationship types**

--- Skip karena tidak ada superclass/subclass --

1. **Many-to-many (\*:\*) binary relationship types**
2. Student dan Course

Student (Banner\_No, FName, LName, Street, City, Postcode, PhoneNumber, Email, DOB, Gender, Category, Nationality, Special\_Needs, Comment, AccStatus, Major, Minor, StaffNo, NOK\_Name)

Primary Key : Banner\_No

Alternate Key : PhoneNumber, Email

Foreign Key : Staff\_No references Advisor

NOK\_Name references NOK

Course (Course\_No, Course\_Name, Course\_Instructor, instructorEmail, instructorPhone, roomNo, departmentNo)

Primary Key : Course\_No

Alternate Key : Course\_Instructor, instructorEmail, instructorPhone

New entity for relation

Enroll (Banner\_No, Course\_No)

Primary Key : Banner\_No, Course\_No

Foreign Key : Banner\_No references Student

Course\_No references Course

1. Bedroom dan Leases

Bedroom (Place\_No, Rent)

Primary Key : Place\_No

Leases (Lease\_No, Lease\_Duration, Start\_Date, Leave\_Date, Invoice\_No, Semester, Payment\_duedate, Payment\_date, Payment\_Method, Reminder)

Primary Key : Lease\_No

Alternate Key : Invoice\_No

Foreign Key : Banner\_No references Student

New entity for relation

For (Place\_No, Leases\_No)

Primary Key : Place\_No, Leases\_No

Foreign Key : Place\_No references Bedroom

Leases\_No references Leases

1. **Complex relationship types**

--- Skip karena tidak ada relasi yang kompleks seperti relasi ternary

1. **Multi-valued attributes**

* Attribut Reminder pada Leases

Reminder (Reminder\_No, Reminder\_Count, Leases\_No)

Primary Key : Reminder\_No

Foreign Key : Leases\_No references Leases

**Step 2.1 RESULT**

Reminder (Reminder\_No, Reminder\_Count, Leases\_No)

Primary Key : Reminder\_No

Foreign Key : Leases\_No references Leases

Bedroom (Place\_No, Rent)

Primary Key : Place\_No

Leases (Lease\_No, Lease\_Duration, Start\_Date, Leave\_Date, Invoice\_No, Semester, Payment\_duedate, Payment\_date, Payment\_Method, Reminder)

Primary Key : Lease\_No

Alternate Key : Invoice\_No

Foreign Key : Banner\_No references Student

For (Place\_No, Leases\_No)

Primary Key : Place\_No, Leases\_No

Foreign Key : Place\_No references Bedroom

Leases\_No references Leases

Student (Banner\_No, FName, LName, Street, City, Postcode, PhoneNumber, Email, DOB, Gender, Category, Nationality, Special\_Needs, Comment, AccStatus, Major, Minor, StaffNo, NOK\_Name)

Primary Key : Banner\_No

Alternate Key : PhoneNumber, Email

Foreign Key : Staff\_No references Advisor

NOK\_Name references NOK

Course (Course\_No, Course\_Name, Course\_Instructor, instructorEmail, instructorPhone, roomNo, departmentNo)

Primary Key : Course\_No

Alternate Key : Course\_Instructor, instructorEmail, instructorPhone

Enroll (Banner\_No, Course\_No)

Primary Key : Banner\_No, Course\_No

Foreign Key : Banner\_No references Student

Course\_No references Course

Hall (Hall\_No, Hall\_Name, Telp\_No, Address)

Primary Key : Hall\_No

Alternate Key : Telp\_No

Staff (Staff\_No, FName, LName, Email, Street, City, Postcode, DOB, Gender, Position, Location, Hall\_No)

Primary Key : Staff\_No

Alternate Key : Email

Foreign Key : Hall\_No references Hall

Flat (Flat\_No, Address)

Primary Key : Flat\_No

Inspection (Inspection\_Date, Property\_Status, Comments, Flat\_No, Staff\_No)

Primary Key : Flat\_No

Foreign Key : Flat\_No references Flat

Staff\_No references Staff

Room (Room\_No, Flat\_No, Hall\_No)

Primary Key : Room\_No

Foreign Key : Flat\_No references Flat

Hall\_No references Hall

Advisor (Department, Int\_TelpNo, Email, Staff\_No)

Primary Key : Staff\_No

Foreign Key : Staff\_No references Staff

NOK (NOK\_Name, Relationship, Street, City, Postcode, Telp\_No)

Primary Key : NOK\_name

Alternate Key : Telp\_No

A screenshot of a cell phone

Description automatically generated

**Step 2.2 Validate relations using normalization**

|  |  |
| --- | --- |
| **Reminder** | |
| **UNF** | @Reminder\_No, Reminder\_Count, Leases\_No |
| **1NF** | @Reminder\_No, Reminder\_Count, Leases\_No |
| **2NF** | @Reminder\_No, Reminder\_Count, Leases\_No |
| **3NF** | @Reminder\_No, Reminder\_Count, Leases\_No |
| **Bedroom** | |
| **UNF** | @Place\_No, Rent |
| **1NF** | @Place\_No, Rent |
| **2NF** | @Place\_No, Rent |
| **3NF** | @Place\_No, Rent |
| **Leases** | |
| **UNF** | @Lease\_No, Lease\_Duration, Start\_Date, Leave\_Date, Invoice\_No, Semester, Payment\_duedate, Payment\_date, Payment\_Method, Reminder |
| **1NF** | @Lease\_No, Lease\_Duration, Start\_Date, Leave\_Date, Invoice\_No, Semester, Payment\_duedate, Payment\_date, Payment\_Method, Reminder |
| **2NF** | @Lease\_No, Lease\_Duration, Start\_Date, Leave\_Date, Invoice\_No, Semester, Payment\_duedate, Payment\_date, Payment\_Method, Reminder |
| **3NF** | @Lease\_No, Lease\_Duration, Start\_Date, Leave\_Date, Invoice\_No, Semester, Payment\_duedate, Payment\_date, Payment\_Method, Reminder |
| **For** | |
| **UNF** | @Place\_No, Leases\_No |
| **1NF** | @Place\_No, Leases\_No |
| **2NF** | @Place\_No, Leases\_No |
| **3NF** | @Place\_No, Leases\_No |
| **Student** | |
| **UNF** | @Banner\_No, FName, LName, Street, City, Postcode, PhoneNumber, Email, DOB, Gender, Category, Nationality, Special\_Needs, Comment, AccStatus, Major, Minor, StaffNo, NOK\_Name |
| **1NF** | @Banner\_No, FName, LName, Street, City, Postcode, PhoneNumber, Email, DOB, Gender, Category, Nationality, Special\_Needs, Comment, AccStatus, Major, Minor, StaffNo, NOK\_Name |
| **2NF** | @Banner\_No, FName, LName, Street, City, Postcode, PhoneNumber, Email, DOB, Gender, Category, Nationality, Special\_Needs, Comment, AccStatus, Major, Minor, StaffNo, NOK\_Name |
| **3NF** | @Banner\_No, FName, LName, Street, City, Postcode, PhoneNumber, Email, DOB, Gender, Category, Nationality, Special\_Needs, Comment, AccStatus, Major, Minor, StaffNo, NOK\_Name |
| **Course** | |
| **UNF** | @Course\_No, Course\_Name, Course\_Instructor, instructorEmail, instructorPhone, roomNo, departmentNo |
| **1NF** | @Course\_No, Course\_Name, Course\_Instructor, instructorEmail, instructorPhone, roomNo, departmentNo |
| **2NF** | @Course\_No, Course\_Name, Course\_Instructor, instructorEmail, instructorPhone, roomNo, departmentNo |
| **3NF** | @Course\_No, Course\_Name, Course\_Instructor, instructorEmail, instructorPhone, roomNo, departmentNo |
| **Enroll** | |
| **UNF** | @Banner\_No, Course\_No |
| **1NF** | @Banner\_No, Course\_No |
| **2NF** | @Banner\_No, Course\_No |
| **3NF** | @Banner\_No, Course\_No |
| **Hall** | |
| **UNF** | @Hall\_No, Hall\_Name, Telp\_No, Address |
| **1NF** | @Hall\_No, Hall\_Name, Telp\_No, Address |
| **2NF** | @Hall\_No, Hall\_Name, Telp\_No, Address |
| **3NF** | @Hall\_No, Hall\_Name, Telp\_No, Address |
| **Staff** | |
| **UNF** | @Staff\_No, FName, LName, Email, Street, City, Postcode, DOB, Gender, Position, Location, Hall\_No |
| **1NF** | @Staff\_No, FName, LName, Email, Street, City, Postcode, DOB, Gender, Position, Location, Hall\_No |
| **2NF** | @Staff\_No, FName, LName, Email, Street, City, Postcode, DOB, Gender, Position, Location, Hall\_No |
| **3NF** | @Staff\_No, FName, LName, Email, Street, City, Postcode, DOB, Gender, Position, Location, Hall\_No |
| **Flat** | |
| **UNF** | @Flat\_No, Address |
| **1NF** | @Flat\_No, Address |
| **2NF** | @Flat\_No, Address |
| **3NF** | @Flat\_No, Address |
| **Inspection** | |
| **UNF** | @Inspection\_Date, Property\_Status, Comments, Flat\_No, Staff\_No |
| **1NF** | @Inspection\_Date, Property\_Status, Comments, Flat\_No, Staff\_No |
| **2NF** | @Inspection\_Date, Property\_Status, Comments, Flat\_No, Staff\_No |
| **3NF** | @Inspection\_Date, Property\_Status, Comments, Flat\_No, Staff\_No |
| **Room** | |
| **UNF** | @Room\_No, Flat\_No, Hall\_No |
| **1NF** | @Room\_No, Flat\_No, Hall\_No |
| **2NF** | @Room\_No, Flat\_No, Hall\_No |
| **3NF** | @Room\_No, Flat\_No, Hall\_No |
| **Advisor** | |
| **UNF** | @Department, Int\_TelpNo, Email, Staff\_No |
| **1NF** | @Department, Int\_TelpNo, Email, Staff\_No |
| **2NF** | @Department, Int\_TelpNo, Email, Staff\_No |
| **3NF** | @Department, Int\_TelpNo, Email, Staff\_No |
| **NOK** | |
| **UNF** | @NOK\_Name, Relationship, Street, City, Postcode, Telp\_No |
| **1NF** | @NOK\_Name, Relationship, Street, City, Postcode, Telp\_No |
| **2NF** | @NOK\_Name, Relationship, Street, City, Postcode, Telp\_No |
| **3NF** | @NOK\_Name, Relationship, Street, City, Postcode, Telp\_No |

**Step 2.3 Validate relations against user transaction**

Listed here are some examples of query transactions that should be supported by

the University Accommodation Office database system:

1. Present a report listing the Manager’s name and telephone number for each hall of residence.
2. Present a report listing the names and banner numbers of students with the details of their lease agreements.
3. Display the details of lease agreements that include the summer semester.
4. Display the details of the total rent paid by a given student.
5. Present a report on students who have not paid their invoices by a given date.
6. Display the details of apartment inspections where the property was found to be in an unsatisfactory condition.
7. Present a report of the names and banner numbers of students with their room number and place number in a particular hall of residence.
8. Present a report listing the details of all students currently on the waiting list for accommodation; that is; who were not placed.
9. Display the total number of students in each student category.
10. Present a report of the names and banner numbers for all students who have not supplied details of their next-of-kin.
11. Display the name and internal telephone number of the Adviser for a particular student.
12. Display the minimum, maximum, and average monthly rent for rooms in residence halls.
13. Display the total number of places in each residence hall.
14. Display the staff number, name, age, and current location of all members of the residence staff who are over 60 years old today.

**A screenshot of a cell phone

Description automatically generated**

**Step 2.4 Check integrity constraint**

Reminder (Reminder\_No, Reminder\_Count, Leases\_No)

Primary Key : Reminder\_No

Foreign Key : Leases\_No references Leases ON UPDATE CASCADE ON DELETE CASCADE

Bedroom (Place\_No, Rent)

Primary Key : Place\_No

Leases (Lease\_No, Lease\_Duration, Start\_Date, Leave\_Date, Invoice\_No, Semester, Payment\_duedate, Payment\_date, Payment\_Method, Reminder)

Primary Key : Lease\_No

Alternate Key : Invoice\_No

Foreign Key : Banner\_No references Student ON UPDATE CASCADE

For (Place\_No, Leases\_No)

Primary Key : Place\_No, Leases\_No

Foreign Key : Place\_No references Bedroom ON UPDATE CASCADE ON DELETE SET NULL

Leases\_No references Leases ON UPDATE CASCADE ON DELETE SET NULL

Student (Banner\_No, FName, LName, Street, City, Postcode, PhoneNumber, Email, DOB, Gender, Category, Nationality, Special\_Needs, Comment, AccStatus, Major, Minor, StaffNo, NOK\_Name)

Primary Key : Banner\_No

Alternate Key : PhoneNumber, Email

Foreign Key : Staff\_No references Advisor ON UPDATE CASCADE ON DELETE SET NULL

NOK\_Name references NOK ON UPDATE CASCADE ON DELETE SET NULL

Course (Course\_No, Course\_Name, Course\_Instructor, instructorEmail, instructorPhone, roomNo, departmentNo)

Primary Key : Course\_No

Alternate Key : Course\_Instructor, instructorEmail, instructorPhone

Enroll (Banner\_No, Course\_No)

Primary Key : Banner\_No, Course\_No

Foreign Key : Banner\_No references Student

Course\_No references Course ON UPDATE CASCADE

Hall (Hall\_No, Hall\_Name, Telp\_No, Address)

Primary Key : Hall\_No

Alternate Key : Telp\_No

Staff (Staff\_No, FName, LName, Email, Street, City, Postcode, DOB, Gender, Position, Location, Hall\_No)

Primary Key : Staff\_No

Alternate Key : Email

Foreign Key : Hall\_No references Hall ON UPDATE CASCADE ON DELETE SET NULL

Flat (Flat\_No, Address)

Primary Key : Flat\_No

Inspection (Inspection\_Date, Property\_Status, Comments, Flat\_No, Staff\_No)

Primary Key : Flat\_No

Foreign Key : Flat\_No references Flat

Staff\_No references Staff ON UPDATE CASCADE ON DELETE CASCADE

Room (Room\_No, Flat\_No, Hall\_No)

Primary Key : Room\_No

Foreign Key : Flat\_No references Flat

Hall\_No references Hall ON UPDATE CASCADE ON DELETE SET NULL

Advisor (Department, Int\_TelpNo, Email, Staff\_No)

Primary Key : Staff\_No

Foreign Key : Staff\_No references Staff ON UPDATE CASCADE ON DELETE SET NULL

NOK (NOK\_Name, Relationship, Street, City, Postcode, Telp\_No)

Primary Key : NOK\_name

Alternate Key : Telp\_No

**Step 2.5 Review logical data model with user**

Pada tahap ini dilakukan review oleh user mengenai requirement dari data model yang ada. Review ini dapat dilakukan dengan berbagai cara seperti wawancara, kuesioner, atau cara yang lain. Tujuan dari review ini adalah untuk memastikan bahwa logical data model yang dibuat merupakan representasi yang diharapkan oleh perusahaan tersebut. Jika user merasa tidak puas mengenai hasil dari data model ini, maka dilakukan perulangan ke tahap sebelumnya untuk merevisi data model tersebut. Oleh karena logical data model ini dibuat dengan pendekatan sentral (centralized approach), maka langsung kepada tahap final dari logical data methodology, pada step 2.7.

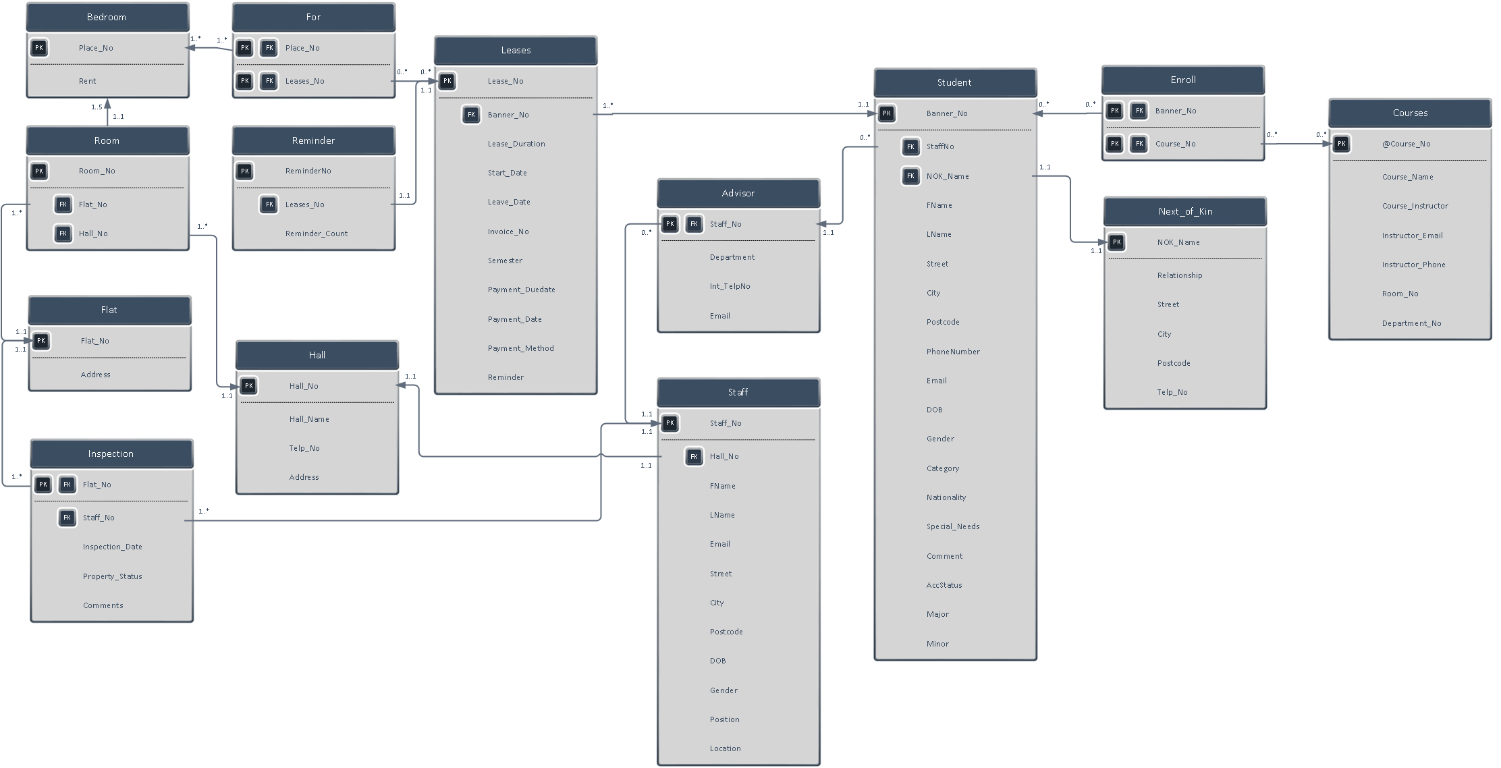
**Step 2.6 Merge logical data models into global data model (optional step)**

(Kita melewati tahapan ini)

**Step 2.7 Check for future growth**

Pada tahap ini, dilakukan pengecekan mengenai perkembangan data model ke depannya. Jika pengecekan secara berkala tidak dilakukan, maka kemungkinan penggunaan data model tersebut akan berakhir dalam waktu yang tidak lama akibat perkembangan data tidak terpenuhi. Maka, tahap ini dilakukan untuk melihat perubahan yang terjadi ke depannya dan menilai apakah logical data model dapat memenuhi perubahan tersebut. Tentunya akan mengeluarkan biaya yang besar jika memenuhi perubahan tersebut setiap terjadi perubahan yang ada. Maka, perubahan tersebut perlu diseleksi seperlunya sehingga perubahan tetap terakomodasi dan biaya yang dikeluarkan dapat diminimalisir.

**Complete ERD**

****

**Physical Database Design**

**Step 3 Translate logical data model for target DBMS**

**Step 3.1 Design base relation**

**Pertanyaan Tambahan :**

1. **Step 2.1 Derive relation for logical data model**
2. Strong entity types

Entitas ini merupakan sebuah entitas yang tidak bergantung pada kehadiran entitas lainnya. Entitas ini dapat berdiri sendiri tanpa adanya kehadiran entitas lain. Biasanya ditandai dengan adanya sebuah primary key (attribute unik).

1. Weak entity types

Entitas ini sangat bergantung dengan keberadaan entitas lainnya. Entitas ini tidak dapat berdiri sendiri karena tidak memiliki atribut yang unik sebagai identitas entitas tersebut. Biasanya ditandai dengan adanya penggunaan foreign key dari entitas lain yang dapat berupa strong entity atau weak entity lainnya.

1. One-to-many (1:\*) binary relationship types

Pada tahapan ini, kita melihat bagian *cardinality* dari setiap *relation* yang ada. Apabila terdapat relasi biner, yang bersifat *one-to-many*(1:\*), maka kita perlu memberikan primary key dari entitas yang memiliki cardinality 1 kepada entitas yang memiliki cardinality *many* (\*) untuk dijadikan primary key atau foreign key.

1. One-to-one (1:1) binary relationship types

Periksa apakah terdapat cardinality yang bersifat *one-to-one* (1:1) pada setiap relation binary yang ada.

1. Mandatory participation on both sides of 1:1 relationship

Apabila terdapat *participation* dari relation tersebut yang bersifat 1:1, maka kita menggabungkan kedua entitas tersebut menjadi sebuah entitas baru. Kita dapat memilih salah satu dari beberapa primary key yang ada dalam salah satu entitas untuk dijadikan primary key pada tabel baru. Untuk key yang tidak terpakai lainnya akan dijadikan alternate key.

1. Mandatory participation on one side of a 1:1 relationship

Apabila salah satu *participation* dari sebuah relasi bersifat *optional*(0), maka entitas nya dijadikan sebagai entitas *parent*. Sedangkan entitas yang memiliki *participation*  *mandatory* (1), maka akan menjadi *child* *entity*. Maka, primary key yang terdapat pada entitas *parent,* akan disalin ke dalam *child entity.*

1. Optional participation on both sides of a 1:1 relationship

Apabila kedua patisipasi dari sebuah relasi adalah optional (0), maka kita pilih salah satu dari kedua entitas dari kedua entias yang ada untuk dijadikan entitas *parent.* Entitas lainnya dijadikan sebagai entitas *child*. Sama seperti sebelumya, primary key yang terdapat pada entitas *parent* akan dicopy ke dalam entitas *child.*

1. One-to-one (1:1) recursive relationship

Relasi rekursif merupakan suatu relasi dimana entitas tersebut menunjuk pada dirinya sendiri.

1. With mandatory participation on both side

Terdapat 2 copy of primary key yang digunakan sebagai penunjuk bahwa terdapat hubungan rekursif.

1. With mandatory participation on one side

Kita dapat membuat single relation yang memiliki 2 copy dari primary key atau membuat relasi baru. Relasi baru tersebut memiliki 2 copy dari primary key. Salinan tersebut sebagai foreign key dan harus diubah Namanya untuk menghindari ambiguitas.

1. With optional participation on both side

Buatlah relasi baru yang memiliki 2 copy dari primary key dimana salinannya dijadikan sebagai foreign key.

1. Superclass/subclass relationship types

Apabila dalam data model kita terdapat *relation superclass/subclass*, kita bisa mengikuti aturan seperti dibawah ini:

* Participation : mandatory , Disjoin constraint : Nondisjoin (and)

Buatlah menjadi sebuah relasi baru

* Participation : optional , Disjoin constraint : Nondisjoin (and)

Buatlah menjadi 2 relasi, dimana terdapat 1 entitas untuk superclass, dan entitas lainnya untuk subclass.

* Participation : mandatory , Disjoin constraint : Disjoin (or)

Kita dapat menjadikannya menjadi banyak relasi, dimana untuk setiap gabungan superclass dan subclass memiliki 1 relasi.

* Participation : optional , Disjoin constraint : Disjoin (or)

Kita menjadikannya banyak relasi, namun 1 relasi untuk superclass dan 1 relasi untuk subclass.

1. Many-to-many (\*:\*) binary relationship types

Jika ada *relation biner* yang bersifat *many-to-many,*  buatlah sebuah relasi yang menghubungkan atribut yang menjadi bagian dari *relation* itu. Semua primary key yang ada pada kedua entitas yang terhubung, disalin dan dijadikan foreign key pada *relation* baru.

1. Complex relationship types

Jika terdapat *relation* selain yang bersifat biner (seperti ternary), buatlah sebuah relation baru yang digunakan untuk melambangkan *relationship* yang ada. Sehingga tidak ada *relationship* yang bersifat kompleks.

1. Multi-valued attributes

Attribut yang biasanya memiliki lebih dari 1 *value* dan dilambangkan dengan [ ], kita perlu membuat suatu *relation* baru untuk mewakili attribute tersebut. Sehingga kita tidak perlu memasukan *multivalue attribute* ke dalam entitas kita.

1. **Step 2.4 Check integrity constraint**
2. Required data

Tentukan apakah suatu attribute bisa tidak memiliki value atau harus memiliki value setiap saat. Apabila attribute bisa saja tidak memiliki nilai, maka dapat dikatakan bahwa nilai dari attribute tersebut adalah NULL.

Ex: FullName VARCHAR (50) NOT NULL

1. Attribute domain constraint

Setiap attribute memiliki domain yang dapat berupa *string, integer, Boolean,* atau lainnya sesuai dengan kebutuhan attribute tersebut.

Ex: StaffID CHAR (5), StaffName VARCHAR (50), DOB DATE

1. Multiplicity

Kita perlu memastikan kembali apakah *multiplicity* yang terdapat dalam setiap relasi sudah sesuai dengan proses bisnis yang dijalankan.

1. Entity integrity

Pastikan bahwa setiap primary key yang ada harus memiliki nilai dan tidak dapat menampung nilai NULL. Karena primary key yang memiliki keunikan berperan sebagai identifikator suatu entitas.

Ex: CustomerID CHAR (5) NOT NULL PRIMARY KEY

1. Referential integrity

Setiap *foreign key* yang terdapat dalam suatu entitas harus merusuk pada *primary key* entitas lainnya. Dan karena pada point D dinyatakan bahwa primary key tidak dapat kosong, maka f*oreign key* yang terdapat dalam suatu entitas tersebut juga tidak dapat kosong. Dan apabila terjadi perubahan nilai pada *primary key*, kita perlu menentukan apakah semua *foreign key* yang terkait dengan *primary key* tersebut ikut berubah atau tidak.

Ex: OrderID int NOT NULL PRIMARY KEY,

OrderNumber int NOT NULL,

PersonID int FOREIGN KEY REFERENCES Persons (PersonID)

1. General constraint

Disini kita memastikan general constraint yang apakah sudah sesuai dengan proses bisnis yang ada.

Ex: Age int CHECK (Age>=18)

**Physical Database Design**

**Step 3 Translate logical data model for target DBMS**

**Step 3.1 Design base relation**

Domain ReminderNumber: variable length character string, length 5

Domain ReminderCount: integer, in range 0 - 2

Domain LeasesNumber variable length character string, length 5

Reminder (

Reminder\_No ReminderNumber NOT NULL,

Reminder\_Count ReminderCount NOT NULL DEFAULT 0,

Leases\_No LeasesNumber NOT NULL,

PRIMARY KEY (Reminder\_No),

FOREIGN KEY (Leases\_No) REFERENCES Leases(Leases\_No) ON UPDATE CASCADE ON DELETE CASCADE);

Domain PlaceNumber variable length character string, length 5

Domain RentRate monetary value

Bedroom (

Place\_No PlaceNumber NOT NULL,

Rent RentRate NOT NULL,

PRIMARY KEY (Place\_No));

Domain LeaseNumber variable length character string, length 5

Domain LeaseDuration integer, in range 1-2

Domain StartDate date value

Domain EndDate date value

Domain InvoiceNumber variable length character string, length 5

Domain Semester integer, in range 1-10

Domain PaymentDueDate date value

Domain PaymentDate date value

Domain PaymentMethod variable length character string, length 15

Domain ReminderNumber variable length character string, length 5

Leases (

Lease\_No LeaseNumber NOT NULL,

Lease\_Duration LeaseDuration NOT NULL,

Start\_Date StartDate NOT NULL,

Leave\_Date EndDate NOT NULL,

Invoice\_No InvoiceNumber NOT NULL,

Semester Semester NOT NULL,

Payment\_duedate PaymentDueDate NOT NULL,

Payment\_date PaymentDate NOT NULL,

Payment\_Method PaymentMethod NOT NULL,

Reminder ReminderNumber NOT NULL,

PRIMARY KEY (Lease\_No),

ALTERNATE KEY (Invoice\_No),

FOREIGN KEY (Banner\_No) REFERENCES Student(Banner\_No) ON UPDATE CASCADE,

FOREIGN KEY (Reminder) REFERENCES Reminder(Reminder\_No) ON UPDATE CASCADE );

Domain PlaceNumber variable length character string, length 5

Domain LeaseNumber variable length character string, length 5

For (

Place\_No PlaceNumber NOT NULL,

Leases\_No LeaseNumber NOT NULL,

PRIMARY KEY (Place\_No, Leases\_No),

FOREIGN KEY (Place\_No) REFERENCES Bedroom(Place\_No) ON UPDATE CASCADE ON DELETE SET NULL,

FOREIGN KEY (Leases\_No) REFERENCES Leases(Leases\_No) ON UPDATE CASCADE ON DELETE SET NULL);

Domain BannerNumber variable length character string, length 5

Domain FirstName variable length character string, length 20

Domain LastName variable length character string, length 20

Domain Street variable length character string, length 25

Domain City variable length character string, length 15

Domain Postcode variable length character string, length 7

Domain PhoneNumber variable length character string, length 15

Domain Email variable length character string, length 30

Domain DOB date value

Domain Gender single character, must be one of ‘M’ or ‘F’

Domain Category variable length character string, length 15

Domain Nationality variable length character string, length 15

Domain SpecialNeeds variable length character string, length 25

Domain Comment variable length character string, length 50

Domain AccStatus variable length character string, length 15

Domain Major variable length character string, length 15

Domain Minor variable length character string, length 15

Domain StaffNo variable length character string, length 5

Domain NOKName variable length character string, length 30

Domain CourseNumber variable length character string, length 5

Student (

Banner\_No BannerNumber NOT NULL,

FName FirstName NOT NULL,

LName LastName NOT NULL,

Street Street NOT NULL,

City City NOT NULL,

Postcode Postcode,

PhoneNumber PhoneNumber NOT NULL,

Email Email NOT NULL,

DOB DOB NOT NULL,

Gender Gender NOT NULL,

Category Category NOT NULL,

Nationality Nationality NOT NULL,

Special\_Needs SpecialNeeds,

Comment Comment,

AccStatus AccStatus NOT NULL,

Major Major NOT NULL,

Minor Minor NOT NULL,

StaffNo StaffNumber NOT NULL,

NOK\_Name NOKName

Course\_No CourseNumber NOT NULL,

PRIMARY KEY (Banner\_No)

ALTERNATE KEY (PhoneNumber, Email)

FOREIGN KEY (Staff\_No) REFERENCES Advisor(Staff\_No) ON UPDATE CASCADE ON DELETE SET NULL

FOREIGN KEY (NOK\_Name) REFERENCES NOK(NOK\_Name) ON UPDATE CASCADE ON DELETE SET NULL

FOREIGN KEY (Course\_No) REFERENCES Course(Course\_No) ON UPDATE CASCADE ON DELETE SET NULL);

Domain CourseNumber variable length character string, length 5

Domain CourseName variable length character string, length 25

Domain InstructorName variable length character string, length 30

Domain InstructorEmail variable length character string, length 30

Domain InstructorPhone variable length character string, length 17

Domain RoomNumber variable length character string, length 5

Domain DepartmentNumber variable length character string, length 5

Course (

Course\_No CourseNumber NOT NULL,

Course\_Name CourseName NOT NULL,

Course\_Instructor InstructorName,

instructorEmail InstructorEmail,

instructorPhone InstructorPhone,

roomNo RoomNumber,

departmentNo DepartmentNumber NOT NULL

PRIMARY KEY (Course\_No)

ALTERNATE KEY (Course\_Instructor, instructorEmail, instructorPhone));

Domain BannerNumber variable length character string, length 5

Domain CourseNumber variable length character string, length 5

Enroll (

Banner\_No BannerNumber NOT NULL,

Course\_No CourseNumber NOT NULL,

PRIMARY KEY (Banner\_No, Course\_No),

FOREIGN KEY (Banner\_No) REFERENCES Student(Banner\_No) ON UPDATE CASCADE ON DELETE CASCADE,

FOREIGN KEY (Course\_No) REFERENCES Course(Course\_No) ON UPDATE CASCADE ON DELETE CASCADE);

Domain HallNumber variable length character string, length 5

Domain HallName variable length character string, length 15

Domain TelphoneNumber variable length character string, length 15

Domain Address variable length character string, length 50

Hall (

Hall\_No HallNumber NOT NULL,

HallName HallName NOT NULL,

Telp\_No TelphoneNumber NOT NULL,

Address Address NOT NULL,

PRIMARY KEY (Hall\_No),

ALTER NATE KEY (Telp\_No));

Domain StaffNumber variable length character string, length 5

Domain FirstName variable length character string, length 20

Domain LastName variable length character string, length 20

Domain Street variable length character string, length 25

Domain City variable length character string, length 15

Domain Postcode variable length character string, length 7

Domain Email variable length character string, length 30

Domain DOB date value

Domain Gender single character, must be one of ‘M’ or ‘F’

Domain Position variable length character string, length 20

Domain Location variable length character string, length 15

Domain HallNumber variable length character string, length 5

Staff (

Staff\_No StaffNumber NOT NULL,

FName FirstName NOT NULL,

LName LastName NOT NULL,

Email Email NOT NULL,

Street Street NOT NULL,

City City NOT NULL,

Postcode Postcode,

DOB DOB NOT NULL,

Gender Gender NOT NULL,

Position Position NOT NULL,

Location Location NOT NULL,

Hall\_No HallNumber,

PRIMARY KEY (Staff\_No),

ALTERNATE KEY (Email)

FOREIGN KEY (Hall\_No) REFERENCES Hall(Hall\_No) ON UPDATE CASCADE ON DELETE SET NULL);

Domain FlatNumber variable length character string, length 5

Domain Address variable length character string, length 50

Flat (

Flat\_No FlatNumber NOT NULL,

Address Address NOT NULL,

PRIMARY KEY (Flat\_No));

Domain InspectionDate date value

Domain PropertyStatus variable length character string, length 20

Domain Comments variable length character string, length 50

Domain FlatNumber variable length character string, length 5

Domain StaffNumber variable length character string, length 5

Inspection (

Inspection\_Date InspectionDate NOT NULL,

Property\_Status PropertyStatus NOT NULL,

Comments Comments NOT NULL,

Flat\_No FlatNumber NOT NULL,

StaffNo StaffNumber NOT NULL,

PRIMARY KEY (Flat\_No)

FOREIGN KEY (Flat\_No) REFERENCES Flat(Flat\_No) ON UPDATE CASCADE

FOREIGN KEY (Staf\_No) REFERENCES Staff(Staf\_No) ON UPDATE CASCADE ON DELETE CASCADE);

Domain RoomNumber variable length character string, length 5

Domain FlatNumber variable length character string, length 5

Domain HallNumber variable length character string, length 5

Room (

Room\_No RoomNumber NOT NULL,

Flat\_No FlatNumber,

Hall\_No HallNumber,

PRIMARY K EY (Room\_No)

FOREIGN KEY (Flat\_No) REFERENCES Flat(Flat\_No) ON UPDATE CASCADE ON DELETE SET NULL,

FOREIGN KEY (Hall\_No) REFERENCES Hall(Hall\_No) ON UPDATE CASCADE ON DELETE SET NULL);

Domain Department variable length character string, length 20

Domain internalTelphoneNumber variable length character string, length 15

Domain Email variable length character string, length 30

Domain StaffNumber variable length character string, length 5

Advisor (

Department Department NOT NULL,

Int\_TelpNo internalTelphoneNumber NOT NULL,

Email Email NOT NULL,

Staff\_No StaffNumber,

PRIMARY KEY (Staff\_No),

FOREIGN KEY (Staff\_No) REFERENCES Staff(Staff\_No) ON UPDATE CASCADE ON DELETE CASCADE);

Domain NOKName variable length character string, length 30

Domain Relationship variable length character string, length 15

Domain Street variable length character string, length 25

Domain City variable length character string, length 15

Domain Postcode variable length character string, length 7

Domain TelphoneNumber variable length character string, length 15

NOK (

NOK\_Name NOKName NOT NULL,

Relationship Relationship NOT NULL,

Street Street NOT NULL,

City City NOT NULL,

Postcode Postcode,

Telp\_No TelphoneNumber NOT NULL,

PRIMARY KEY (NOK\_Name),

ALTERNATE KEY (Telp\_No));

**Step 3.2 Design representation of derived data**

Staff

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Staff\_No | FName | LName | Gender | Position | Location | TotalofInspection |
| S0001 | Andi | Budi | M | Hall Manager | Hall A | 2 |
| S0002 | Budi | Cahyadi | M | Hall Assistant | Hall A | 1 |
| S0003 | Cahyadi | Darmawan | M | Cleaner | Hall B | 0 |
| S004 | Aulia | Mahendra | F | Advisor | University | 0 |

Inspection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Flat\_No | Staff\_No | Inspection\_Date | Property\_Status | Comments |
| F0001 | S0001 | 2-05-2020 | Satisfying |  |
| F0002 | S0002 | 16-04-2020 | Not Satisfying |  |
| F0003 | S0001 | 29-03-2020 | Not Satisfying |  |

Hall

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Hall\_No | Hall\_Name | Telp\_No | Address | TotalStaff | Totalroom |
| H0001 | Hall A | 021- 123 456 | Jl. Anggrek | 2 | 1 |
| H0002 | Hall B | 021- 789 123 | Jl. Syahdan | 1 | 1 |

Flat

|  |  |  |
| --- | --- | --- |
| Flat\_No | Address | TotalRoom |
| F0001 | Jl. Manggis | 1 |

Room

|  |  |  |
| --- | --- | --- |
| Room\_No | Flat\_No | Hall\_No |
| R0001 | F0001 |  |
| R0002 |  | H002 |
| R0003 |  | H001 |

Student

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Banner\_No | Staff\_No | FName | LName | Gender | Course\_No |
| B0001 | S0004 | Ani | Natalika | F | C0001 |
| B0002 | S0004 | Budi | Yuli | M | C0001 |
| B0003 | S0004 | Caca | Andica | F | C0002 |

Advisor

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Staff\_No | Department | Int\_TelpNo | Email | TotalStudent |
| S0004 | SOCS | 61233 | au@email.com | 3 |

Courses

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Course\_No | CourseName | CourseInstructor | RoomNo | DepartmentNo |
| C0001 | Algorithm | Axel Tan | 101 | D0001 |
| C0002 | Database | Frans Lim | 102 | D0002 |

**Step 3.3 Design general constraints**

* One flat for maximum 5 person

CONSTRAINT MaxFlatPerson

CHECK ( NOT EXIST (

SELECT Flat\_No, COUNT(BannerID) AS “totalPerson”

FROM Room r JOIN Bedroom b ON r.Room\_No = b.Room\_No

JOIN Lease l ON l. Place\_No = b.Place\_No

GROUP BY Flat\_No

HAVING COUNT(BannerID) >5 )

)

**Step 4 Design file organizations and indexes**

**Step 4.1 Analyze transactions**

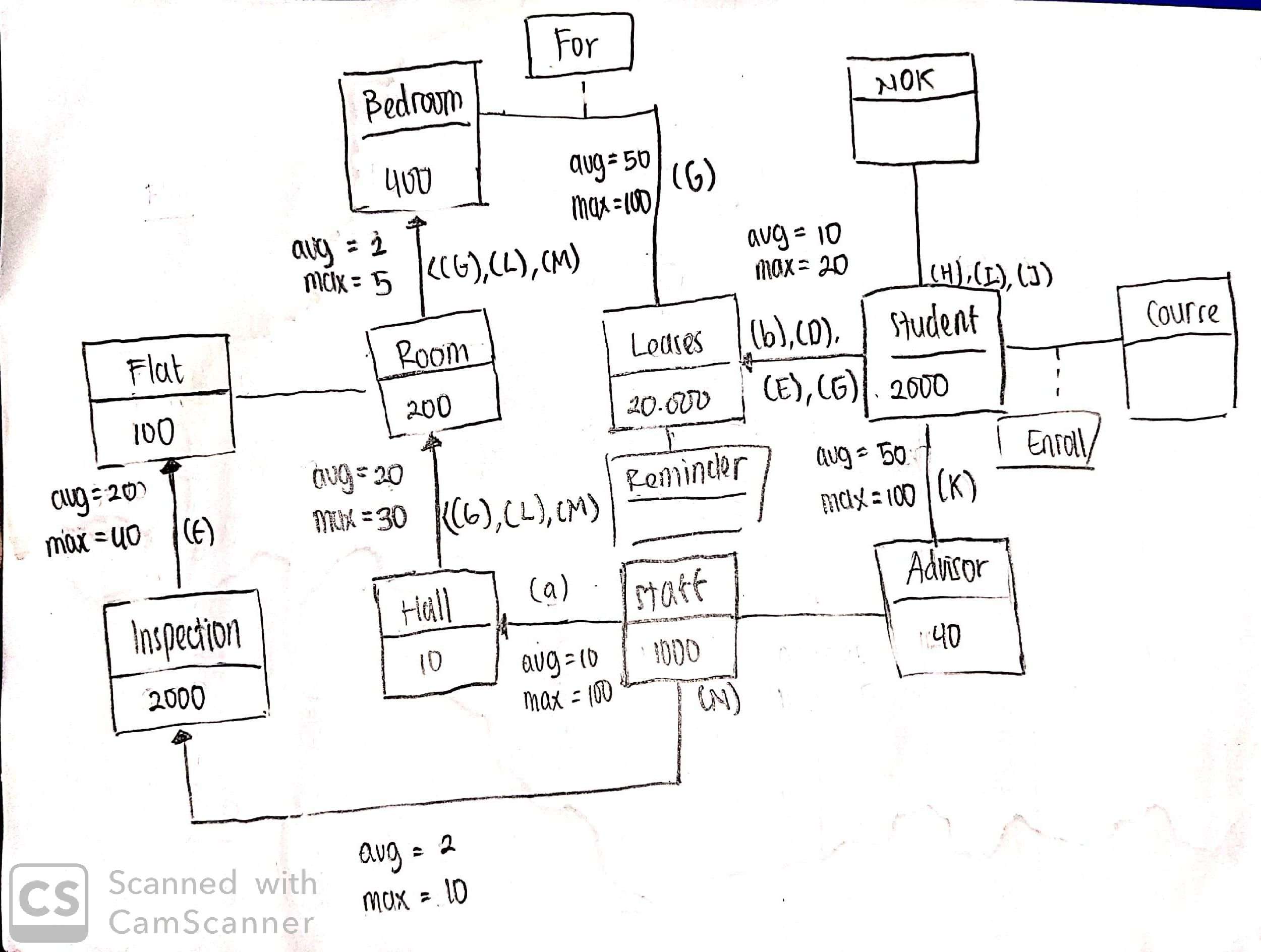
1. Present a report listing the Manager’s name and telephone number for each hall of residence.
2. Present a report listing the names and banner numbers of students with the details of their lease agreements.
3. Display the details of lease agreements that include the summer semester.
4. Display the details of the total rent paid by a given student.
5. Present a report on students who have not paid their invoices by a given date.
6. Display the details of apartment inspections where the property was found to be in an unsatisfactory condition.
7. Present a report of the names and banner numbers of students with their room number and place number in a particular hall of residence.
8. Present a report listing the details of all students currently on the waiting list for accommodation; that is; who were not placed.
9. Display the total number of students in each student category.
10. Present a report of the names and banner numbers for all students who have not supplied details of their next-of-kin.
11. Display the name and internal telephone number of the Adviser for a particular student.
12. Display the minimum, maximum, and average monthly rent for rooms in residence halls.
13. Display the total number of places in each residence hall.
14. Display the staff number, name, age, and current location of all members of the residence staff who are over 60 years old today

Dari contoh query transaction dari a-n, berikut ini adalah tabel hasil cross referencing dengan relasi yang ada.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Relation** | **A** | | | | **B** | | | | **C** | | | | **D** | | | | **E** | | | |
| **I** | **R** | **U** | **D** | **I** | **R** | **U** | **D** | **I** | **R** | **U** | **D** | **I** | **R** | **U** | **D** | **I** | **R** | **U** | **D** |
| Reminder |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Bedroom |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Leases |  |  |  |  |  | V |  |  |  | V |  |  |  | V |  |  |  | V |  |  |
| For |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Student |  |  |  |  |  | V |  |  |  |  |  |  |  | V |  |  |  | V |  |  |
| Course |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Enroll |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Hall |  | V |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Staff |  | V |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Flat |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Inspection |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Room |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Advisor |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| NOK |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Relation** | **F** | | | | **G** | | | | **H** | | | | **I** | | | | **J** | | | |
| **I** | **R** | **U** | **D** | **I** | **R** | **U** | **D** | **I** | **R** | **U** | **D** | **I** | **R** | **U** | **D** | **I** | **R** | **U** | **D** |
| Reminder |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Bedroom |  |  |  |  |  | V |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Leases |  |  |  |  |  | V |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| For |  |  |  |  |  | V |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Student |  |  |  |  |  | V |  |  |  | V |  |  |  | V |  |  |  | V |  |  |
| Course |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Enroll |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Hall |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Staff |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Flat | V |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Inspection | V |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Room |  |  |  |  |  | V |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Advisor |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| NOK |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Relation** | **K** | | | | **L** | | | | **M** | | | | **N** | | | |
| **I** | **R** | **U** | **D** | **I** | **R** | **U** | **D** | **I** | **R** | **U** | **D** | **I** | **R** | **U** | **D** |
| Reminder |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Bedroom |  |  |  |  |  | V |  |  |  |  |  |  |  |  |  | 2 |
| Leases |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 5 |
| For |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Student |  | V |  |  |  |  |  |  |  |  |  |  |  |  |  | 8 |
| Course |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Enroll |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Hall |  |  |  |  |  | V |  |  |  | V |  |  |  |  |  | 3 |
| Staff |  | V |  |  |  |  |  |  |  |  |  |  |  | V |  | 3 |
| Flat |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |
| Inspection |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |
| Room |  |  |  |  |  | V |  |  |  | V |  |  |  |  |  | 3 |
| Advisor |  | V |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |
| NOK |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |



**Step 4.2 Choose file organizations**

Dalam kasus ini, kami memilih untuk menggunakan clusters sebagai file organization. Karena dengan cluster, kita dapat dengan mudah mengelompokan data berdasarkan label.

**Step 4.3 Choose indexes**

Primary indexes

* CREATE INDEX Staff\_No ON Staff(StaffNo) CLUSTER;
* CREATE INDEX Banner\_No ON Student(Banner\_No) CLUSTER;
* CREATE INDEX Lease\_No ON Lease(Lease\_No) CLUSTER;

Secondary Indexes

* CREATE INDEX Banner\_No ON Lease(Banner\_No);
* CREATE INDEX StaffPosition ON Staff(Position);
* CREATE INDEX LeaseSemester ON Lease(Semester);
* CREATE INDEX Condition ON Inspect(Comments);

|  |  |
| --- | --- |
| Table | Index |
| Staff | Staff\_No |
|  | StaffPosition |
| Student | Banner\_No |
| Lease | Lease\_No |
|  | Banner\_No |
|  | LeaseSemester |
| Inspect | Condition |

**Step 4.4 Estimate disk space requirements**

**#**Dijawab pada bagian pertanyaan tambahan#

**Step 5 Design user views**

* Staff consist of Hall Manager, Hall Assistant, Adviser, Administrative Assistant user views
* Student consist of Student, Adviser, And Administrative Assistant user view
* Flat consist of Administrative Assistant user view
* Hall consist of Hall Manager and Hall Assistant user view

**Step 6 Design security mechanism**

Hall Manager

GRANT SELECT ON Staff, Hall TO Hall Manager

Administrative Assistant

GRANT SELECT, INSERT, UPDATE, DELETE ON Staff, Student, Flat TO Administrative Assistant

Hall Assistant

GRANT SELECT, INSERT, UPDATE, DELETE ON Hall TO Hall Assistant

Adviser

GRANT SELECT ON Staff, Student to Adviser

Student

GRANT SELECT, UPDATE ON Student to Student

**Pertanyaan Tambahan**

1. Tujuan physical database design

Physical database design merupakan suatu tahapan dalam database design yang dilakukan setelah logical database design. Tujuan utamanya adalah agar semua yang telah di desain pada tahapan conceptual dan logical dapat diterapkan pada DBMS yang tersedia. Dlaam tahapan physical, kita harus tau bagaimana target DBMS bekerja dan bagaimana caranya kita dapat mengimplementasikan logical yang sudah disusun ke dalam DBMS tersebut

1. Langkah-langkah dari physical database design

Step 3 Translate logical data model for target DBMS

Pada tahapan ini, kita membuat skema dari relational database yang dikembangkan dari logical database. Tujuannya agar semua yang telah diperkirakan dalam logical database dapat diterapkan secara langsung pada target DBMS.

Step 3.1 Design base relation

Pada tahapan ini kita menentukan domain apa saja yang diperlukan oleh setiap attribute yang terdapat dalam setiap relation yang ada. Domain brisikan mengenai tipe data yang ditampung, panjang maksimal yang dapat ditampung, dan atau spesifik data yang dapat ditampung. Selain itu juga kita perlu menentukan apakah atribut tersebut dapat menampung data yang bersifat NULL atau tidak.

Step 3.2 Design representation of derived data

Dalam tahapan ini, kita perlu menentukan apakah suatu derived data perlu dimasukan ke dalam database kita atau tidak. Penentuan tersebut bergantung pada kebutuhan dari data tersebut. Kita perlu memperhatikan bahwa semakin banyak derived data yang disimpan, maka akan semakin besar juga biaya penyimpanan yang dilakukan.

Step 3.3 Design general constraints

Disini kita dapat menentukan batasan-batasan umum yang ada dengan tujuan untuk membatasi jumlah transaksi yang dilakukan terhadap suatu relasi.

Step 4 Design file organizations and indexes

Pada step 4 bertujuan untuk menentukan file organization dan index yang dapat mengoptimasi database yang hendak dibuat.

Step 4.1 Analyze transactions

Kita perlu menganalisa semua kemungkinann transaksi yang terjadi beserta dengan frekuensi akses terhadap relasi tersebut. Disini kita melihat apa saja table yang sering terlibat dalam transaksi dan jenis transaksi apa saja yang dilakukan.

Step 4.2 Choose file organizations

File organization berperan sangat penting dalam mengoptimasi suatu database yang hendak didesain. Namun tidak semua jenis file organization didukung oleh DBMS yang bersangkutan. Oleh karena itu kita perlu memilihnya dengan bijak.

Step 4.3 Choose indexes

Adanya penambahan index terhadap table yang sering diakses akan meningkatkan performa dari database tersebut. Namun terdapat beberapa hal yang perlu diperhatikan dalam pembuatan index. Penentuan index yang salah dapat membuat pemrosesan transaksi yang ada lebih lambat dari yang sebelumnya.

Step 4.4 Estimate disk space requirements

Setelah semua step diatas dilakukan, kita dapat menentukan seberapa besar penyimpanna yang perlu disediakan untuk menampung secara keseluruhan database yang dibangun.

Step 5 Design user views

Dalam tahapan ini kita menentukan kembali user view untuk setiap table yang terlibat dalam setiap transaksi yang ada.

Step 6 Design security mechanism

Dalam step ini, kita membuat suatu sistem keamanan dimana tidak setiap user dapat mengakses semua table yang ada dalam database. Kita dapat memberikan akses (GRANT) atau menolak akses (REVOKE) berdasarkan user view yang ada. Sehingga data-data yang tersimpan dalam database akan tetap terintegrasi dan akurat.

1. Estimasi disk space untuk kasus diatas

