158.337 Project Marking Sheet (Part A – 17.5% course mark)

(Attach this to your project report BEFORE you turn it in.)
(Please make sure you provide all the necessary details)

Oracle Account: Group 19

Group Member 1:

ID number: 22009303

Name: Meixian Shi

Group Member 2:

ID number: 22009300

Name: Xiaotong Chen

Group Member 3:

ID number: 22009144

Name: Zhongyu Zhang

(Grader's section, please do not write below this)

PART A:	/60 marks
Logical Database Design (Step 1):	/30 marks
Physical Database Design (Step 2 and 3):	/30 marks

Contents

1 Normalization Process

1.1 1st NF to 2nd NF

Student Table Normalization

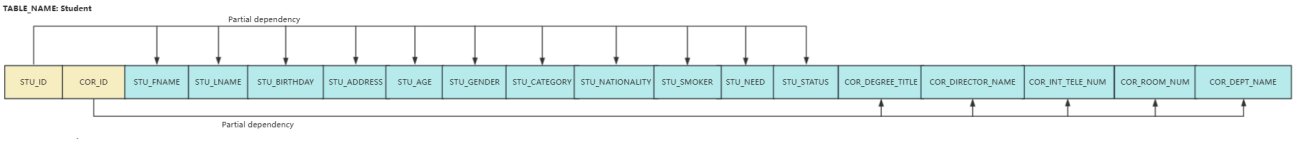

Figure 1: Student Table(before)

The `Student` table has the following attributes:

STU_ID,	COR_ID,
STU_FNAME,	STU_LNAME,
STU_BIRTHDAY,	STU_ADDRESS,
STU_AGE,	STU_GENDER,
STU_CATEGORY,	STU_NATIONALITY,
STU_SMOKER,	STU_STATUS,
COR_DEGREE_TITLE,	COR_INT_TELE_NUM,
COR_ROOM_NUM,	COR_DEPT_NAME

The following partial dependencies exist in the `Student` table:

- STU_ID → { STU_FNAME, STU_LNAME, STU_BIRTHDAY, STU_ADDRESS, STU_AGE, STU_GENDER, STU_CATEGORY, STU_NATIONALITY, STU_SMOKER, STU_STATUS }
- COR_ID → { COR_DEGREE_TITLE, COR_DIRECTOR_NAME, COR_INT_TELE_NUM, COR_ROOM_NUM, COR_DEPT_NAME }

To achieve 2NF, we need to create new tables to eliminate these partial dependencies. This can be done by separating the attributes based on their dependencies:

After removing partial dependencies, the following tables are created:
Student Table

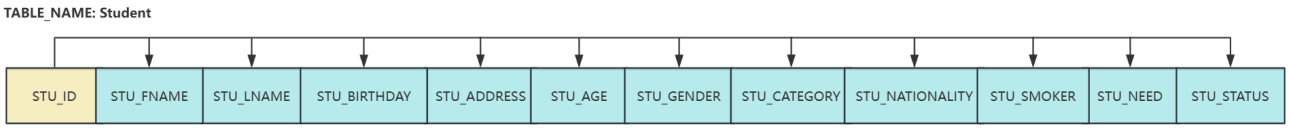

Figure 2: Student Table(after)

The `Student` table now has the following attributes:

STU_ID,	STU_FNAME,
STU_LNAME,	STU_BIRTHDAY,
STU_ADDRESS,	STU_AGE,
STU_GENDER,	STU_CATEGORY,
STU_NATIONALITY,	STU_SMOKER,
STU_STATUS	

Course Table

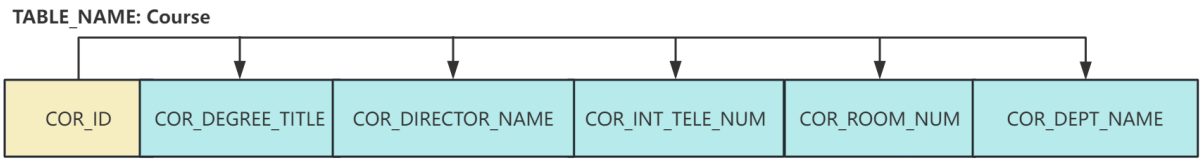

Figure 3: Course Table

The `Course` table now has the following attributes:

`COR_ID`, `COR_DEGREE_TITLE`,
`COR_DIRECTOR_NAME`, `COR_INT_TELE_NUM`,
`COR_ROOM_NUM`, `COR_DEPT_NAME`

Hall Table Normalization

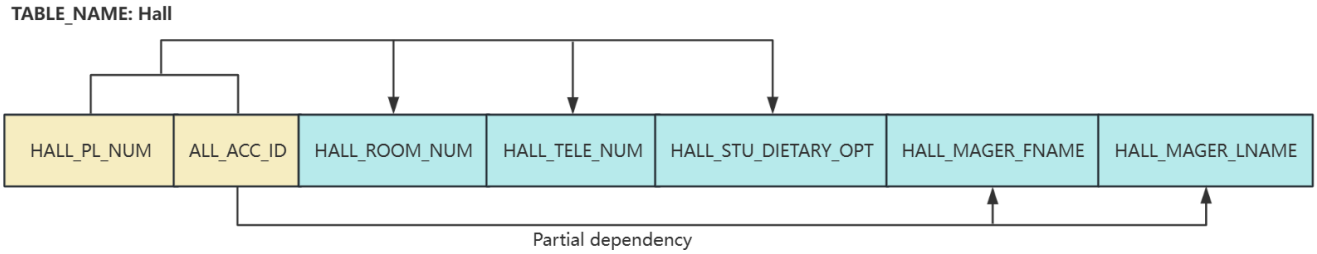

Figure 4: Hall Table(before)

The `Hall` table has the following attributes:

`HALL_PL_NUM`, `ALL_ACC_ID`,
`HALL_ROOM_NUM`, `HALL_TELE_NUM`,
`HALL_STU_DIETARY_OPT`, `HALL_MAGER_FNAME`,
`HALL_MAGER_LNAME`

The following partial dependencies exist in the `Hall` table:
`ALL_ACC_ID` $\rightarrow$ `(HALL_MAGER_FNAME, HALL_MAGER_LNAME)`

To achieve 2NF, we need to create new tables to eliminate the partial dependency.

After removing partial dependencies, the following tables are created:

Hall Table

The `Hall` table now has the following attributes:

`HALL_PL_NUM`, `ALL_ACC_ID`,
`HALL_ROOM_NUM`, `HALL_TELE_NUM`,
`HALL_STU_DIETARY_OPT`

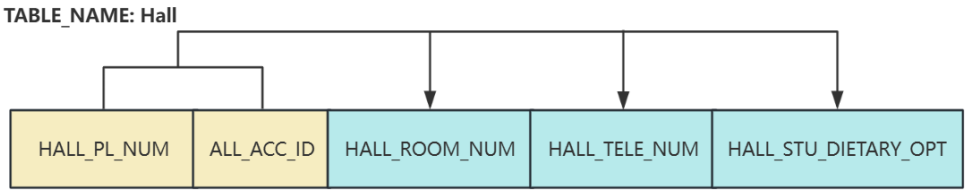

Figure 5: Hall Table(after)

Manager Table

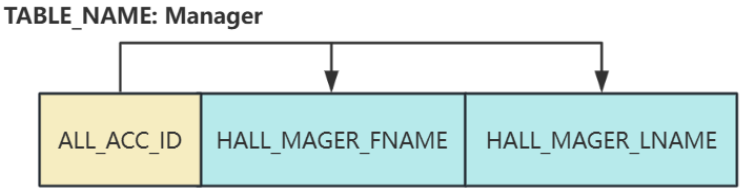

Figure 6: Manager Table

The `Manager` table now has the following attributes:

`ALL_ACC_ID`, `HALL_MAGER_FNAME`, `HALL_MAGER_LNAME`

1.2 2nd NF to 3th NF

Invoice Table Normalization

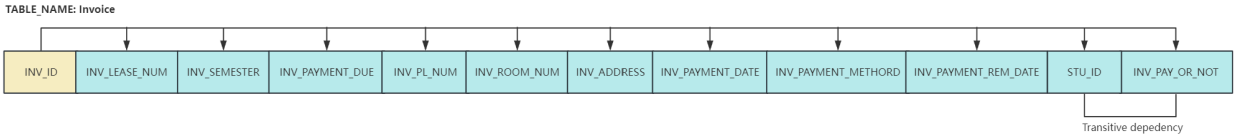

Figure 7: Invoice Table(before)

The `Invoice` table has the following attributes:

`INV_ID`, `INV_LEASE_NUM`, `INV_SEMESTER`, `INV_PAYMENT_DUE`, `INV_PL_NUM`, `INV_ROOM_NUM`, `INV_ADDRESS`, `INV_PAYMENT_DATE`, `INV_PAYMENT_METHORD`, `INV_PAYMENT_REM_DATE`, `STU_ID`, `INV_PAY_OR_NOT`

The following transitive dependency exists in the `Invoice` table:
`INV_ID` →`STU_ID` → `INV_PAY_OR_NOT`

To achieve 3NF, we need to create new tables to eliminate the transitive dependency.

After removing the transitive dependency, the following tables are created:
Invoice Table

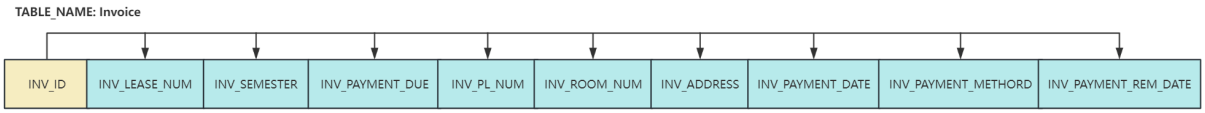

Figure 8: Invoice Table(after)

The `Invoice` table now has the following attributes:

`INV_ID`,	`INV_LEASE_NUM`,
`INV_SEMESTER`,	`INV_PAYMENT_DUE`,
`INV_PL_NUM`,	`INV_ROOM_NUM`,
`INV_ADDRESS`,	`INV_PAYMENT_DATE`,
`INV_PAYMENT_METHORD`,	`INV_PAYMENT_REM_DATE`

PaymentList Table

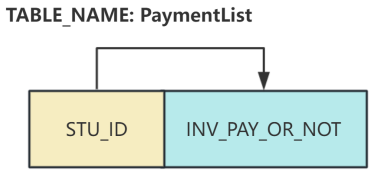

Figure 9: PaymentList Table

The `PaymentList` table now has the following attributes:

`STU_ID`, `INV_PAY_OR_NOT`

1.3 BCNF to 4nd NF

Advisor Table Normalization

The `Advisor` table has the following attributes:

`ADVISOR_ID`,	`ADVISOR_FNAME`,
`ADVISOR_LNAME`,	`ADVISOR_BIRTHDAY`,
`ADVISOR_POSITION`,	`ADVISOR_DEPARTMENT`,
`ADVISOR_TELE_NUM`,	`ADVISOR_ROOM_NUM`

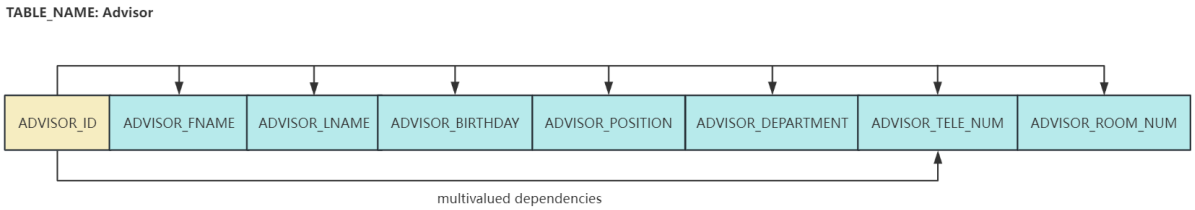

Figure 10: Advisor Table(before)

The following multivalued dependencies exist in the `Advisor` table:

- `ADVISOR_ID` → `ADVISOR_TELE_NUM`

To achieve 4NF, we need to create new tables to eliminate these multivalued dependencies. This can be done by separating the attributes based on their dependencies:

After removing multivalued dependencies, the following tables are created:
Advisor Table

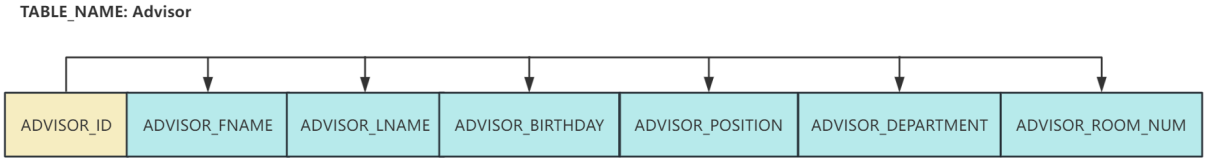

Figure 11: Advisor Table(after)

The `Advisor` table now has the following attributes:

`ADVISOR_ID`, `ADVISOR_FNAME`,
`ADVISOR_LNAME`, `ADVISOR_BIRTHDAY`,
`ADVISOR_POSITION`, `ADVISOR_DEPARTMENT`,
`ADVISOR_ROOM_NUM`

AdvisorTele Table

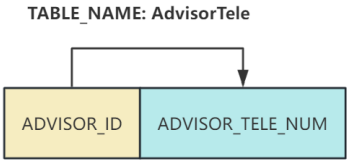

Figure 12: AdvisorTele Table

The `AdvisorTele` table now has the following attributes:

`ADVISOR_ID`, `ADVISOR_TELE_NUM`

Next_Of_Kin Table Normalization

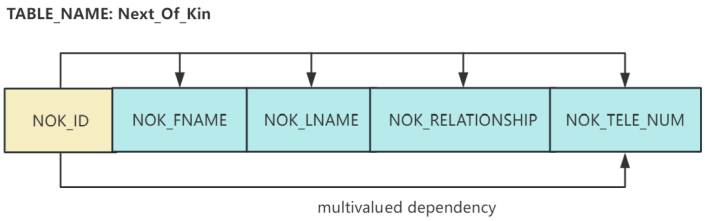

Figure 13: Next-Of-Kin Table(before)

The `Next_Of_Kin` table has the following attributes:

`NOK_ID`, `NOK_FNAME`,
`NOK_LNAME`, `NOK_RELATIONSHIP`,
`NOK_TELE_NUM`

The following multivalued dependency exists in the `Next_Of_Kin` table:

- `NOK_ID` → `NOK_TELE_NUM`

To achieve 4NF, we need to create new tables to eliminate this multivalued dependency. This can be done by separating the attributes based on their dependencies:

After removing multivalued dependencies, the following tables are created:
Next_Of_Kin Table

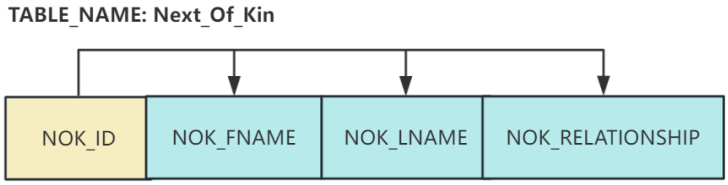

Figure 14: Next-Of-Kin Table(after)

The `Next_Of_Kin` table now has the following attributes:

`NOK_ID`, `NOK_FNAME`,
`NOK_LNAME`, `NOK_RELATIONSHIP`

NOK_Tele Table

The `NOK_Tele` table now has the following attributes:

`NOK_ID`, `NOK_TELE_NUM`

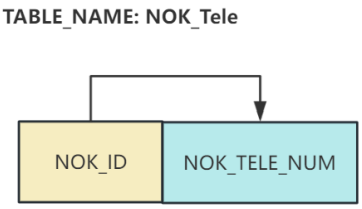

Figure 15: NOK_Tele Table

2 ER Diagram

2.1 Final 4th NF ERD

2.1.1 Detailed ERD Version

Notice:

1. This version of the ERD includes all the essential information, such as the data types, sizes, primary keys, foreign keys, relationship details, strength (identifying or non-identifying), and participation (mandatory or optional). While this comprehensive view ensures that every aspect is covered, it may appear complex.

2. In case the ERD images in this PDF are not clear, a link to the original Visio file has been provided on GitHub. You can access the source file here for better clarity.
Click here to access GitHub repository

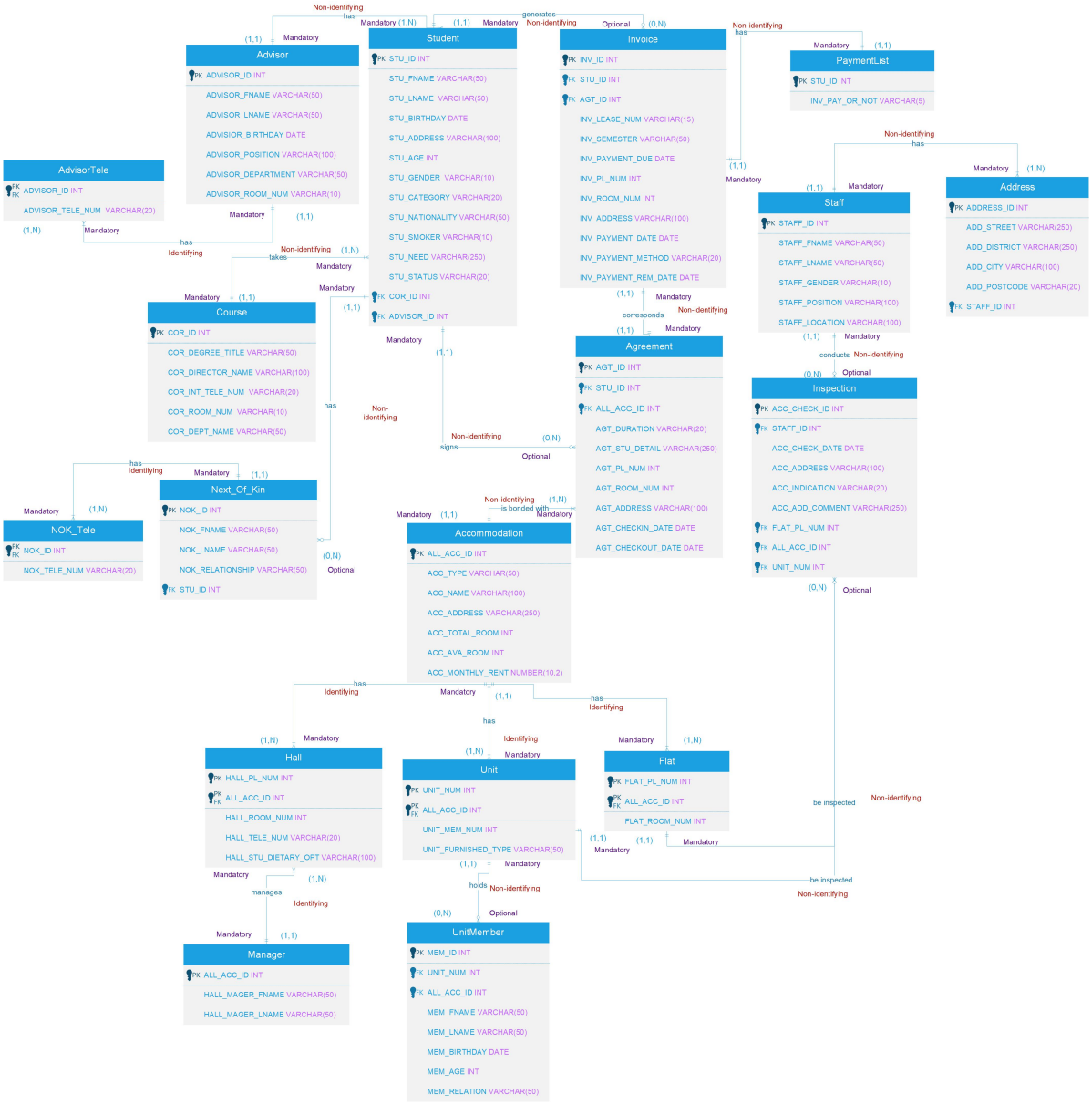

Figure 16: Detailed ERD Version

2.1.2 Simplified ERD Version

Notice:

1. This version of the ERD omits some of the detailed information, such as data types, sizes, and relationship specifics, to provide a clearer and more accessible visual representation. While it does not include every detail, it offers an easier-to-read version.

2. GitHub link for source file:
Click here to access GitHub repository

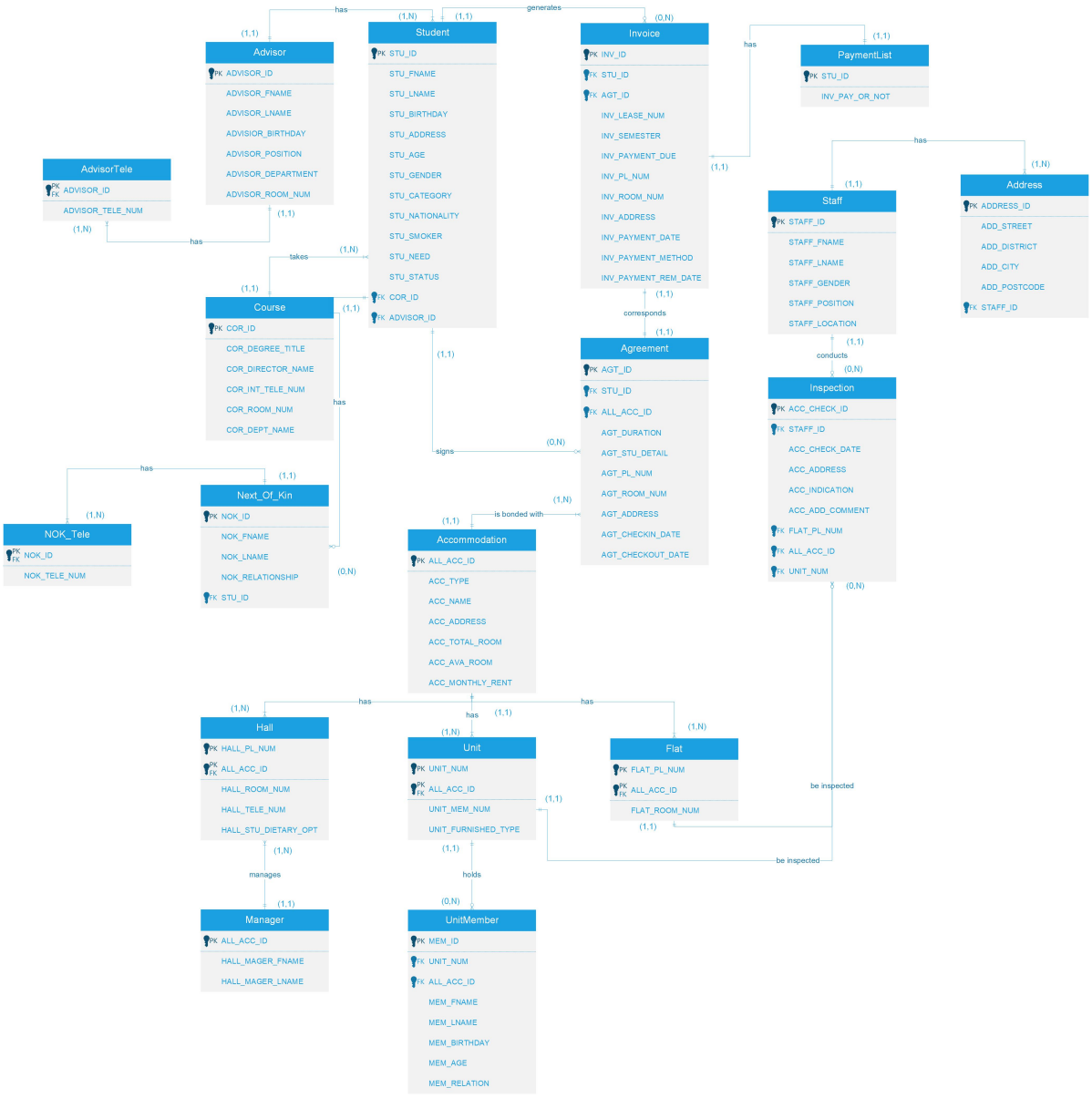

Figure 17: Simplified ERD Version

3 Assumptions

The following assumptions are made to simplify problems.

Assumptions 1: Each student may have zero or more Next-Of-Kin contacts, but each Next-Of-Kin corresponds to only one student, and there are no duplicate students in the entire accommodation system.

Assumptions 2: The `ALL_ACC_ID` attribute represents the type of accommodation, with each type corresponding to a unique number in the range of 1-7. Specifically, 1-3 represent Hall-type accommodations, where 1 stands for Hall1 (Tui), 2 stands for Hall2 (Pukeko), and 3 stands for Hall3 (Weka). Numbers 4-6 represent Flat-type accommodations: 4 stands for Flat1 (Titoki), 5 stands for Flat2 (Matipo), and 6 stands for Flat3 (Tanekaha). Number 7 represents Unit (Tanekaha).

Assumptions3: The range for HALL_ROOM_NUM is 1-70, the HALL_PL_NUM is composed of the ALL_ACC_ID as the hundreds digit, plus the corresponding HALL_ROOM_NUM.

Assumptions4: For Flat1 (Titoki) and Flat2 (Matipo), the FLAT_ROOM_NUM ranges from 1-20. For Flat3 (Tanekaha), the FLAT_ROOM_NUM ranges from 1-30. The range for FLAT_PL_NUM is composed of the ALL_ACC_ID as the hundreds digit, plus the corresponding FLAT_ROOM_NUM.

Assumptions5: The range for UNIT_NUM is 1-12.

Assumptions6: A student with a STU_STATUS of "waiting" means they have applied for accommodation, but the school has not yet approved it. A student with a STU_STATUS of "placed" means their accommodation application has been approved, and they are currently renting school accommodation.

Assumptions7: Only students with a STU_STATUS of "placed" will have an Agreement and Invoice.

Assumptions8: Advisors and Next-Of-Kin may have one or more phone numbers.

Assumptions9: In a Unit, recorded Members do not include the student themselves, only the student's partner and/or children, meaning there may be 0-2 Members.

Assumptions10: The academic year has three semesters: semester 1, semester 2, and the summer semester. Semester 1 runs from March 1 to June 30 each year; semester 2 runs from July 1 to October 31; the summer semester runs from November 1 to February 29 (or 28).

Notice:

1. To clarify assumptions 2-5, specific examples are provided below:

Accommodation	ALL_ACC_ID	Range
Hall1 (Tui)	1	HALL_PL_NUM: 101-170, HALL_ROOM_NUM: 1-70
Hall2 (Pukeko)	2	HALL_PL_NUM: 201-270, HALL_ROOM_NUM: 1-70
Hall3 (Weka)	3	HALL_PL_NUM: 301-370, HALL_ROOM_NUM: 1-70
Flat1 (Titoki)	4	FLAT_PL_NUM: 401-420, FLAT_ROOM_NUM: 1-20
Flat2 (Matipo)	5	FLAT_PL_NUM: 501-520, FLAT_ROOM_NUM: 1-20
Flat3 (Tanekaha)	6	FLAT_PL_NUM: 601-630, FLAT_ROOM_NUM: 1-30
Unit (Tanekaha)	7	UNIT_NUM: 1-12

Table 1: Accommodation Details

2. Please note that due to our limited understanding of the details of Massey University's accommodation system, there may be some differences between the data in insertData.sql and the actual system. We appreciate your understanding and apologize for any discrepancies.

4 Business Rules

1. Student and Invoice
    - A student may generate zero or multiple invoices.
    - An invoice is generated by only one student.
2. Student and Advisor
    - A student has only one advisor.
    - An advisor can monitor one or more students.
3. Student and Course
    - A student can take only one degree course.
    - A course is taken by one or more students.
4. Student and Next_Of_Kin
    - A student has zero or more next-of-kins.
    - A next-of-kin is only related to one student.
5. Student and Agreement
    - A student can sign zero or more agreements.
    - An agreement is only signed by one student.
6. Advisor and AdvisorTele
    - An advisor can have one or more telephone numbers.
    - A telephone number belongs to only one advisor.
7. Next_Of_Kin and NOK_Tele
    - A next-of-kin can have one or more telephone numbers.
    - A telephone number belongs to only one next-of-kin.
8. Invoice and Agreement
    - An invoice corresponds to one agreement.
    - An agreement is linked to one invoice.
9. Agreement and Accommodation
    - An agreement is bonded with one type of accommodation.
    - A type of accommodation may have one or more agreements.
10. Accommodation and Hall
    - A type of accommodation has one or many different halls.
    - A hall belongs to one type of accommodation.

11. Accommodation and Unit
    - A type of accommodation has one or many different units.
    - A unit belongs to one type of accommodation.
12. Accommodation and Flat
    - A type of accommodation has one or many different flats.
    - A flat belongs to one type of accommodation.
13. Hall and Manager
    - A manager manages one or more halls.
    - A hall is managed by only one manager.
14. Unit and UnitMember
    - A unit can hold zero or many members.
    - A member only belongs to one unit.
15. Invoice and PaymentList
    - An invoice has one payment list.
    - A payment list is associated with one invoice.
16. Staff and Address
    - A staff member can have one or more addresses.
    - An address belongs to only one staff member.
17. Staff and Inspection
    - A staff member may conduct zero or more inspections.
    - An inspection can only be conducted by one staff member.
18. Flat and Inspection
    - A flat can be inspected zero or more times.
    - An inspection is conducted for only one flat.
19. Unit and Inspection
    - A unit can be inspected zero or more times.
    - An inspection is conducted for only one unit.

5 DDL Code

```
    -- Dropping existing tables to avoid conflicts
    DROP TABLE Advisor CASCADE CONSTRAINTS;
    -- Creating the Advisor table
    CREATE TABLE Advisor (
        ADVISOR_ID INT PRIMARY KEY,
        ADVISOR_FNAME VARCHAR(50) NOT NULL,
        ADVISOR_LNAME VARCHAR(50) NOT NULL,
        ADVISOR_BIRTHDAY DATE,
        ADVISOR_POSITION VARCHAR(100),
        ADVISOR_DEPARTMENT VARCHAR(50),
        ADVISOR_ROOM_NUM VARCHAR(10)
    );

    DROP TABLE AdvisorTel CASCADE CONSTRAINTS;
    -- Creating the AdvisorTel table
    CREATE TABLE AdvisorTel (
        ADVISOR_ID INT PRIMARY KEY,
        ADVISOR_TEL_NUM VARCHAR(20) NOT NULL,
        FOREIGN KEY(ADVISOR_ID) REFERENCES Advisor(ADVISOR_ID
  ) ON DELETE CASCADE
    );

    DROP TABLE Accommodation CASCADE CONSTRAINTS;
    -- Creating the Accommodation table
    CREATE TABLE Accommodation (
        ALL_ACC_ID INT PRIMARY KEY,
        ACC_TYPE VARCHAR(50),
        ACC_NAME VARCHAR(100) NOT NULL,
        ACC_ADDRESS VARCHAR(250),
        ACC_TOTAL_ROOM INT,
        ACC_AVA_ROOM INT,
        ACC_MONTHLY_RENT NUMBER(10, 2) NOT NULL
    );

    DROP TABLE Course CASCADE CONSTRAINTS;
    -- Creating the Course table
    CREATE TABLE Course (
        COR_ID INT PRIMARY KEY,
        COR_DEGREE_TITLE VARCHAR(100) NOT NULL,
        COR_DIRECTOR_NAME VARCHAR(100),
        COR_INT_TEL_NUM VARCHAR(20),
        COR_ROOM_NUM VARCHAR(10),
        COR_DEPT_NAME VARCHAR(50)
    );

    DROP TABLE Student CASCADE CONSTRAINTS;
    -- Creating the Student table
    CREATE TABLE Student (
        STU_ID INT PRIMARY KEY,
```

```
        STU_FNAME VARCHAR(50) NOT NULL,
        STU_LNAME VARCHAR(50) NOT NULL,
        STU_BIRTHDAY DATE,
        STU_ADDRESS VARCHAR(100),
        STU_AGE INT NOT NULL,
        STU_GENDER VARCHAR(10),
        STU_CATEGORY VARCHAR(20),
        STU_NATIONALITY VARCHAR(50),
        STU_SMOKER VARCHAR(10),
        STU_NEED VARCHAR(250),
        STU_STATUS VARCHAR2(20),
        COR_ID INT,
        ADVISOR_ID INT,
        FOREIGN KEY(COR_ID) REFERENCES Course(COR_ID),
        FOREIGN KEY(ADVISOR_ID) REFERENCES Advisor
    );

    DROP TABLE Manager CASCADE CONSTRAINTS;
    -- Creating the Manager table
    CREATE TABLE Manager (
        ALL_ACC_ID INT PRIMARY KEY,
        HALL_MAGER_FNAME VARCHAR(50) NOT NULL,
        HALL_MAGER_LNAME VARCHAR(50) NOT NULL,
        FOREIGN KEY(ALL_ACC_ID) REFERENCES Accommodation(
ALL_ACC_ID) ON DELETE CASCADE
    );

    DROP TABLE Staff CASCADE CONSTRAINTS;
    -- Creating the Staff table
    CREATE TABLE Staff (
        STAFF_ID INT PRIMARY KEY,
        STAFF_FNAME VARCHAR(50) NOT NULL,
        STAFF_LNAME VARCHAR(50) NOT NULL,
        STAFF_GENDER VARCHAR(10),
        STAFF_POSITION VARCHAR(100),
        STAFF_LOCATION VARCHAR(100)
    );

    DROP TABLE Address CASCADE CONSTRAINTS;
    -- Creating the Address table
    CREATE TABLE Address (
        ADDRESS_ID INT PRIMARY KEY,
        ADD_STREET VARCHAR(250),
        ADD_DISTRICT VARCHAR(250),
        ADD_CITY VARCHAR(100),
        ADD_POSTCODE VARCHAR(20),
        STAFF_ID INT,
        FOREIGN KEY (STAFF_ID) REFERENCES Staff(STAFF_ID) ON
DELETE CASCADE
    );
```

```
 DROP TABLE Agreement CASCADE CONSTRAINTS;
 -- Creating the Agreement table
 CREATE TABLE Agreement (
     AGT_ID INT PRIMARY KEY,
     STU_ID INT,
     ALL_ACC_ID INT,
     AGT_DURATION VARCHAR(20),
     AGT_STU_DETAIL VARCHAR(250),
     AGT_PL_NUM INT,
     AGT_ROOM_NUM INT,
     AGT_ADDRESS VARCHAR(100),
     AGT_CHECKIN_DATE DATE,
     AGT_CHECKOUT_DATE DATE,
     FOREIGN KEY(STU_ID) REFERENCES Student(STU_ID) ON
DELETE CASCADE,
     FOREIGN KEY(ALL_ACC_ID) REFERENCES Accommodation(
ALL_ACC_ID) ON DELETE CASCADE
 );

 DROP TABLE Hall CASCADE CONSTRAINTS;
 -- Creating the Hall table
 CREATE TABLE Hall (
     HALL_PL_NUM INT,
     ALL_ACC_ID INT,
     HALL_ROOM_NUM INT,
     HALL_TELE_NUM VARCHAR(20),
     HALL_STU_DIETARY_OPT VARCHAR(100),
     PRIMARY KEY(HALL_PL_NUM,ALL_ACC_ID),
     FOREIGN KEY(ALL_ACC_ID) REFERENCES Accommodation ON
DELETE CASCADE
 );

 DROP TABLE Flat CASCADE CONSTRAINTS;
 -- Creating the Flat table
 CREATE TABLE Flat (
     FLAT_PL_NUM INT,
     ALL_ACC_ID INT,
     FLAT_ROOM_NUM INT,
     PRIMARY KEY(FLAT_PL_NUM,ALL_ACC_ID),
     FOREIGN KEY(ALL_ACC_ID) REFERENCES Accommodation ON
DELETE CASCADE
 );

 DROP TABLE Unit CASCADE CONSTRAINTS;
 -- Creating the Unit table
 CREATE TABLE Unit (
     UNIT_NUM INT,
     ALL_ACC_ID INT,
     UNIT_MEM_NUM INT,
     UNIT_FURNISHED_TYPE VARCHAR(50),
     PRIMARY KEY(UNIT_NUM,ALL_ACC_ID),
```

```
        FOREIGN KEY(ALL_ACC_ID) REFERENCES Accommodation ON
DELETE CASCADE
 );

 DROP TABLE UnitMember CASCADE CONSTRAINTS;
 -- Creating the UnitMember table
 CREATE TABLE UnitMember (
        MEM_ID INT PRIMARY KEY,
        UNIT_NUM INT,
        ALL_ACC_ID INT,
        MEM_FNAME VARCHAR(50) NOT NULL,
        MEM_LNAME VARCHAR(50) NOT NULL,
        MEM_AGE INT,
        MEM_RELATION VARCHAR(50),
        FOREIGN KEY (UNIT_NUM, ALL_ACC_ID) REFERENCES Unit (
UNIT_NUM, ALL_ACC_ID)
 );

 DROP TABLE Invoice CASCADE CONSTRAINTS;
 -- Creating the Invoice table
 CREATE TABLE Invoice (
        INV_ID INT PRIMARY KEY,
        STU_ID INT,
        AGT_ID INT,
        INV_LEASE_NUM VARCHAR(15),
        INV_SEMSTER VARCHAR(50),
        INV_PAYMENT_DUE DATE,
        INV_PL_NUM INT,
        INV_ROOM_NUM INT,
        INV_ADDRESS VARCHAR(100),
        INV_PAYMENT_DATE VARCHAR(100),
        INV_PAYMENT_METHOD VARCHAR(20),
        INV_PAYMENT_REM_DATE DATE,
        FOREIGN KEY(STU_ID) REFERENCES Student(STU_ID) ON
DELETE CASCADE,
        FOREIGN KEY(AGT_ID) REFERENCES Agreement(AGT_ID) ON
DELETE CASCADE
 );

 DROP TABLE PaymentList CASCADE CONSTRAINTS;
 -- Creating the PaymentList table
 CREATE TABLE PaymentList (
        STU_ID INT PRIMARY KEY,
        INV_PAY_OR_NOT VARCHAR(5)
 );

 DROP TABLE Next_Of_Kin CASCADE CONSTRAINTS;
 -- Creating the Next_Of_Kin table
 CREATE TABLE Next_Of_Kin (
        NOK_ID INT PRIMARY KEY,
        NOK_FNAME VARCHAR(50) NOT NULL,
```

```
        NOK_LNAME VARCHAR(50) NOT NULL,
        NOK_RELATIONSHIP VARCHAR(50),
        STU_ID INT,
        FOREIGN KEY(STU_ID) REFERENCES Student(STU_ID) ON
    DELETE CASCADE
    );

    DROP TABLE NOK_Tel CASCADE CONSTRAINTS;
    -- Creating the NOK_Tel table
    CREATE TABLE NOK_Tel (
        NOK_ID INT PRIMARY KEY,
        NOK_TEL_NUM VARCHAR(20),
        FOREIGN KEY(NOK_ID) REFERENCES Next_Of_Kin ON DELETE
    CASCADE
    );

    DROP TABLE Inspection CASCADE CONSTRAINTS;
    -- Creating the Inspection table
    CREATE TABLE Inspection (
        ACC_CHECK_ID INT PRIMARY KEY,
        STAFF_ID INT,
        ACC_CHECK_DATE DATE,
        ACC_ADDRESS VARCHAR(100),
        ACC_INDICATION VARCHAR(20),
        ACC_ADD_COMMENT VARCHAR(250),
        FLAT_PL_NUM INT,
        ALL_ACC_ID INT,
        UNIT_NUM INT,
        FOREIGN KEY(STAFF_ID) REFERENCES Staff ON DELETE
    CASCADE,
        FOREIGN KEY(FLAT_PL_NUM, ALL_ACC_ID) REFERENCES Flat
    (FLAT_PL_NUM, ALL_ACC_ID) ON DELETE CASCADE,
        FOREIGN KEY(ALL_ACC_ID) REFERENCES Accommodation (
    ALL_ACC_ID) ON DELETE CASCADE,
        FOREIGN KEY(UNIT_NUM, ALL_ACC_ID) REFERENCES Unit (
    UNIT_NUM, ALL_ACC_ID) ON DELETE CASCADE
    );

    --end of script

    COMMIT;
```

Listing 1: SQL Script for Creating Tables

6 DML Code

```
    -- Insert data into Advisor table
    INSERT INTO Advisor VALUES(1, 'Elizabeth', 'Turner',
    TO_DATE('1970-05-14', 'YYYY-MM-DD'), 'Senior Lecturer', '
    Computer Science', 'CS101');
```

```
 INSERT INTO Advisor VALUES(2, 'Richard', 'Harris',
TO_DATE('1980-07-22', 'YYYY-MM-DD'), 'Professor', '
Engineering', 'EN202');
 INSERT INTO Advisor VALUES(3, 'Laura', 'Mitchell',
TO_DATE('1975-03-11', 'YYYY-MM-DD'), 'Lecturer', '
Mathematics', 'MA303');
 INSERT INTO Advisor VALUES(4, 'Andrew', 'Scott', TO_DATE(
'1965-09-30', 'YYYY-MM-DD'), 'Associate Professor', '
Physics', 'PH404');
 INSERT INTO Advisor VALUES(5, 'Sarah', 'Brown', TO_DATE('
1985-12-17', 'YYYY-MM-DD'), 'Professor', 'Chemistry', '
CH505');

 -- Insert data into AdvisorTel table
 INSERT INTO AdvisorTel VALUES(1, '021-345-6789');
 INSERT INTO AdvisorTel VALUES(2, '021-987-6543');
 INSERT INTO AdvisorTel VALUES(3, '021-456-7890');
 INSERT INTO AdvisorTel VALUES(4, '021-654-1234');
 INSERT INTO AdvisorTel VALUES(5, '021-789-3456');

 -- Insert data into Accommodation table
 INSERT INTO Accommodation VALUES(1, 'Hall', 'Tui', '15
University Rd', 70, 49, 800);
 INSERT INTO Accommodation VALUES(2, 'Hall', 'Pukeko', '27
 Campus Drive', 70, 12, 850);
 INSERT INTO Accommodation VALUES(3, 'Hall', 'Weka', '42
Scholar Ave', 70, 20, 900);
 INSERT INTO Accommodation VALUES(4, 'Flat', 'Titoki', '58
 Lecture Way', 20, 10, 950);
 INSERT INTO Accommodation VALUES(5, 'Flat', 'Matipo', '59
 Research Blvd', 20, 8, 1000);
 INSERT INTO Accommodation VALUES(6, 'Flat', 'Tanekaha', '
98 Research Blvd', 30, 2, 950);
 INSERT INTO Accommodation VALUES(7, 'Unit', 'Tanekaha', '
99 Research Blvd', 12, 1, 1050);

 -- Insert data into Course table
 INSERT INTO Course VALUES(1, 'Bachelor of Computer
Science', 'Elizabeth Turner', '021-345-6789', 'CS101', '
Computer Science');
 INSERT INTO Course VALUES(2, 'Bachelor of Mechanical
Engineering', 'Richard Harris', '021-987-6543', 'EN202', '
Engineering');
 INSERT INTO Course VALUES(3, 'Bachelor of Science in
Mathematics', 'Laura Mitchell', '021-456-7890', 'MA303', '
Mathematics');
 INSERT INTO Course VALUES(4, 'Bachelor of Science in
Physics', 'Andrew Scott', '021-654-1234', 'PH404', '
Physics');
 INSERT INTO Course VALUES(5, 'Bachelor of Science in
Chemistry', 'Sarah Brown', '021-789-3456', 'CH505', '
```

```
Chemistry');

 -- Insert data into Student table
 INSERT INTO Student VALUES(1, 'Alice', 'Johnson', TO_DATE
('2000-01-15', 'YYYY-MM-DD'), '123 Main St', 24, 'Female',
 'Undergraduate', 'New Zealand', 'No', NULL, 'Placed', 1,
1);
 INSERT INTO Student VALUES(2, 'Michael', 'Chen', TO_DATE(
'1999-02-20', 'YYYY-MM-DD'), '456 Elm St', 25, 'Male', '
Postgraduate', 'China', 'No', 'Visual Impairment', 'Placed
', 2, 2);
 INSERT INTO Student VALUES(3, 'Sophia', 'Lee', TO_DATE('
2001-03-25', 'YYYY-MM-DD'), '789 Oak St', 23, 'Female', '
Undergraduate', 'South Korea', 'No', NULL, 'Placed', 3, 3)
;
 INSERT INTO Student VALUES(4, 'Daniel', 'Williams',
TO_DATE('1998-04-30', 'YYYY-MM-DD'), '101 Pine St', 26, '
Male', 'Postgraduate', 'USA', 'Yes', 'Hearing Impairment',
 'Placed', 4, 4);
 INSERT INTO Student VALUES(5, 'Grace', 'Kim', TO_DATE('
1997-05-10', 'YYYY-MM-DD'), '202 Maple St', 27, 'Female',
'Undergraduate', 'New Zealand', 'No', 'Allergic to dust',
'Placed', 5, 5);

 -- Insert data into Manager table
 --Since there are 3 halls, there are only 3 managers here
 INSERT INTO Manager VALUES(1, 'Jonathan', 'Reed');
 INSERT INTO Manager VALUES(2, 'Emily', 'White');
 INSERT INTO Manager VALUES(3, 'Matthew', 'Green');

 -- Insert data into Staff table
 INSERT INTO Staff VALUES(1, 'Jonathan', 'Reed', 'Male', '
Hall Manager', 'Tui Hall');
 INSERT INTO Staff VALUES(2, 'Emily', 'White', 'Female', '
Hall Manager', 'Pukeko Hall');
 INSERT INTO Staff VALUES(3, 'Matthew', 'Green', 'Male', '
Cleaner', 'Weka Hall');
 INSERT INTO Staff VALUES(4, 'Natalie', 'Black', 'Female',
 'Administrative Assistant', 'Main Office');
 INSERT INTO Staff VALUES(5, 'Christopher', 'King', 'Male'
, 'Maintenance Supervisor', 'Campus Maintenance');

 -- Insert data into Address table
 INSERT INTO Address VALUES(1, '15 University Rd', 'Albany
', 'Auckland', '0632', 1);
 INSERT INTO Address VALUES(2, '27 Campus Drive', 'Albany'
, 'Auckland', '0632', 2);
 INSERT INTO Address VALUES(3, '42 Scholar Ave', 'Albany',
 'Auckland', '0632', 3);
 INSERT INTO Address VALUES(4, '58 Lecture Way', 'Albany',
 'Auckland', '0632', 4);
```

```
    INSERT INTO Address VALUES(5, '99 Research Blvd', 'Albany
', 'Auckland', '0632', 5);

    -- Insert data into Agreement table
    INSERT INTO Agreement VALUES(1, 1, 1, '10 Months', 'Alice
    Johnson', 101, 1, '15 University Rd', TO_DATE('2024-02-01
', 'YYYY-MM-DD'), TO_DATE('2024-11-30', 'YYYY-MM-DD'));
    INSERT INTO Agreement VALUES(2, 2, 2, '6 Months', '
Michael Chen', 201, 1, '27 Campus Drive', TO_DATE('
2024-05-01', 'YYYY-MM-DD'), TO_DATE('2024-10-31', 'YYYY-MM
-DD'));
    INSERT INTO Agreement VALUES(3, 3, 3, '12 Months', '
Sophia Lee', 301, 1, '42 Scholar Ave', TO_DATE('2024-02-01
', 'YYYY-MM-DD'), TO_DATE('2025-1-31', 'YYYY-MM-DD'));
    INSERT INTO Agreement VALUES(4, 4, 6, '5 Months', 'Daniel
    Williams', 401, 1, '98 Research Blvd', TO_DATE('
2024-07-01', 'YYYY-MM-DD'), TO_DATE('2024-11-30', 'YYYY-MM
-DD'));
    INSERT INTO Agreement VALUES(5, 5, 7, '12 Months', 'Grace
    Kim', NULL, 1, '99 Research Blvd', TO_DATE('2024-02-01',
'YYYY-MM-DD'), TO_DATE('2025-1-31', 'YYYY-MM-DD'));


    -- Insert data into Hall table
    INSERT INTO Hall VALUES(101, 1, 1, '021-123-4567', '
Vegetarian');
    INSERT INTO Hall VALUES(201, 2, 1, '021-234-5678', '
Vegetarian');
    INSERT INTO Hall VALUES(301, 3, 1, '021-345-6789', NULL);
    INSERT INTO Hall VALUES(102, 1, 2, '021-123-4567', NULL);
    INSERT INTO Hall VALUES(302, 3, 2, '021-345-6789', '
Vegetarian');


    -- Insert data into Flat table
    INSERT INTO Flat VALUES(401, 4, 1);
    INSERT INTO Flat VALUES(501, 5, 1);
    INSERT INTO Flat VALUES(601, 6, 1);
    INSERT INTO Flat VALUES(402, 4, 2);
    INSERT INTO Flat VALUES(502, 5, 20);

    -- Insert data into Unit table
    INSERT INTO Unit VALUES(1, 7, 0, 'partly');
    INSERT INTO Unit VALUES(2, 7, 1, 'partly');
    INSERT INTO Unit VALUES(3, 7, 0, 'partly');
    INSERT INTO Unit VALUES(4, 7, 2, 'fully');
    INSERT INTO Unit VALUES(5, 7, 2, 'fully');

    -- Insert data into UnitMember table
    INSERT INTO UnitMember VALUES(1, 2, 7, 'Linda', 'Li', 25,
    'partner');
```

```
 INSERT INTO UnitMember VALUES(2, 4, 7, 'May', 'Carter',
29, 'partner');
 INSERT INTO UnitMember VALUES(3, 4, 7, 'Frank', 'Williams
', 5, 'child');
 INSERT INTO UnitMember VALUES(4, 5, 7, 'Susan', 'Wright',
 28, 'partner');
 INSERT INTO UnitMember VALUES(5, 5, 7, 'Mark', 'Wright',
3, 'child');


 -- Insert data into Invoice table
 INSERT INTO Invoice VALUES(1, 1, 1, 'INV2024-001', 'S1,
S2', TO_DATE('2024-01-15', 'YYYY-MM-DD'), 101, 1, '15
University Rd', TO_DATE('2024-01-01', 'YYYY-MM-DD'), '
Credit Card', TO_DATE('2024-01-01', 'YYYY-MM-DD'));
 INSERT INTO Invoice VALUES(2, 2, 2, 'INV2024-002', 'S1,
S2', TO_DATE('2024-04-15', 'YYYY-MM-DD'), 201, 1, '27
Campus Drive', TO_DATE('2024-02-28', 'YYYY-MM-DD'), 'Bank
Transfer', NULL);
 INSERT INTO Invoice VALUES(3, 3, 3, 'INV2024-003', 'S1,
S2, SUMMER', TO_DATE('2024-01-15', 'YYYY-MM-DD'), 301, 1,
'42 Scholar Ave', TO_DATE('2024-01-10', 'YYYY-MM-DD'), '
Cheque', TO_DATE('2024-01-01', 'YYYY-MM-DD'));
 INSERT INTO Invoice VALUES(4, 4, 4, 'INV2024-004', 'S2',
TO_DATE('2024-06-15', 'YYYY-MM-DD'), 401, 1, '98 Research
Blvd', TO_DATE('2024-04-13', 'YYYY-MM-DD'), 'Credit Card',
 NULL);
 INSERT INTO Invoice VALUES(5, 5, 5, 'INV2024-005', 'S1,
S2, SUMMER', TO_DATE('2024-01-15', 'YYYY-MM-DD'), NULL, 1,
 '99 Research Blvd', NULL, 'Credit Card', TO_DATE('
2024-01-01', 'YYYY-MM-DD'));


 -- Insert data into PaymentList table
 INSERT INTO PaymentList VALUES(1, 'YES');
 INSERT INTO PaymentList VALUES(2, 'NO');
 INSERT INTO PaymentList VALUES(3, 'YES');
 INSERT INTO PaymentList VALUES(4, 'NO');
 INSERT INTO PaymentList VALUES(5, 'YES');

 -- Insert data into Next_Of_Kin table
 INSERT INTO Next_Of_Kin VALUES(1, 'Mary', 'Johnson', '
Mother', 1);
 INSERT INTO Next_Of_Kin VALUES(2, 'John', 'Chen', 'Father
', 2);
 INSERT INTO Next_Of_Kin VALUES(3, 'Jane', 'Lee', 'Mother'
, 3);
 INSERT INTO Next_Of_Kin VALUES(4, 'Robert', 'Williams', '
Father', 4);
 INSERT INTO Next_Of_Kin VALUES(5, 'Anna', 'Kim', 'Mother'
, 5);
```

```
-- Insert data into NOK_Tel table
INSERT INTO NOK_Tel VALUES(1, '021-567-1234');
INSERT INTO NOK_Tel VALUES(2, '021-890-4567');
INSERT INTO NOK_Tel VALUES(3, '021-234-7890');
INSERT INTO NOK_Tel VALUES(4, '021-345-6789');
INSERT INTO NOK_Tel VALUES(5, '021-567-8901');

-- Insert data into Inspection table
INSERT INTO Inspection VALUES(1, 1, TO_DATE('2024-03-15',
'YYYY-MM-DD'), '98 Research Blvd', 'Satisfactory', 'All
good', 601, 6, NULL);
INSERT INTO Inspection VALUES(2, 2, TO_DATE('2024-03-20',
'YYYY-MM-DD'), '58 Lecture Way', 'Satisfactory', 'Minor
issue with heating', 401, 4, NULL);
INSERT INTO Inspection VALUES(3, 3, TO_DATE('2024-03-25',
'YYYY-MM-DD'), '99 Research Blvd', 'Unsatisfactory', '
Broken window', NULL, 7, 1);
INSERT INTO Inspection VALUES(4, 4, TO_DATE('2024-03-30',
'YYYY-MM-DD'), '58 Lecture Way', 'Satisfactory', 'No
issues found', 401, 4, NULL);
INSERT INTO Inspection VALUES(5, 5, TO_DATE('2024-04-05',
'YYYY-MM-DD'), '59 Research Blvd', 'Unsatisfactory', '
Leaky faucet', 501, 5, NULL);


--end of script

COMMIT;
```

Listing 2: SQL Script for Inserting Data

7 Test Data

The test data for this report is available in the GitHub repository. Due to the large volume of data, it is not practical to include the full datasets directly within the report. Instead, we have provided links to the data files stored in the repository, where you can find the specific tables in the ‘testData‘ directory. For detailed information and to download the files, please visit the following link:

Click here to access GitHub repository

Once on the GitHub page, navigate to the ‘database(exported)‘ and then ‘testData‘ directory to access the tables and other relevant data.