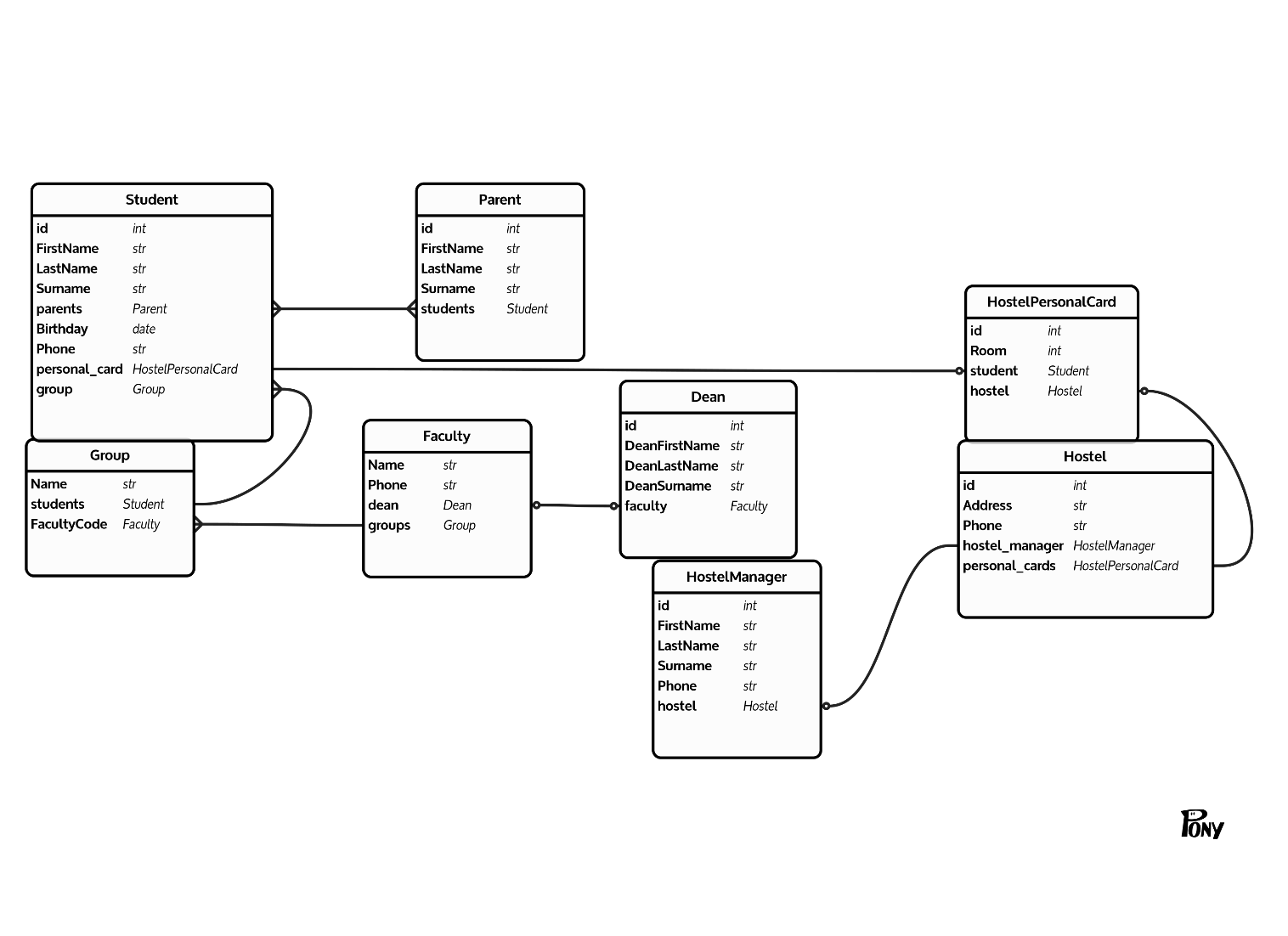
Лабораторна робота №1

Концептуальне моделювання БД

Мета: набуття практичних навичок щодо побудови концептуальних моделей баз даних із використанням діаграмних технологій (модель сутність-зв‘язок).

Хід виконання

Завдання: Всі студенти інституту, яким потрібен гуртожиток реєструються у студмістечку. При поселенні в гуртожиток на студента заводиться ОСОБИСТА КАРТКА, у якій зберігається така інформація: номер картки, ПІБ, дата народження, факультет, група, № гуртожитки, № кімнати, адреса батьків. У студмістечку є довідники: ФАКУЛЬТЕТИ, де для кожного факультету зберігається назва факультету, телефон деканату, ПІБ декана; ГРУПИ, де зазначені шифр групи і код факультету; ГУРТОЖИТКИ, де зазначений номер гуртожитку, адреса, телефон, ПІБ, коменданта.

Рис. 1. Загальна схема бази даних

Таблиця Parent

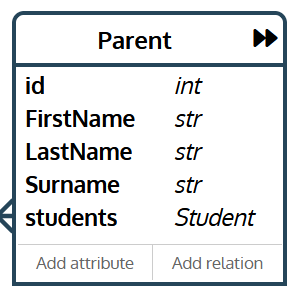


Рис. 2. Таблиця Parent

Ця таблиця призначення для збереження інформації про батьків студента. Зв’язок між студент батьки один до багатьох: в одного студента може бути багато батьків. Таблиця полів та типів:

| **Field Name** | **DataType** | **PK** | **FK** | **Description** |
| --- | --- | --- | --- | --- |
| id | int | + | - | Айді одного з батьків студента |
| FirstName | Str(32) | - | - | Ім’я |
| LastName | Str(32) | - | - | Прізвище |
| Surname | Str(32) | - | - | По батькові, не обов’язкове |
| Students | int | - | + | Айді студента |

SQL код створення таблиці:

CREATE TABLE `parent` (

`id` INTEGER PRIMARY KEY AUTO\_INCREMENT,

`firstname` VARCHAR(32) NOT NULL,

`lastname` VARCHAR(32) NOT NULL,

`surname` VARCHAR(32)

);

Таблиця Student

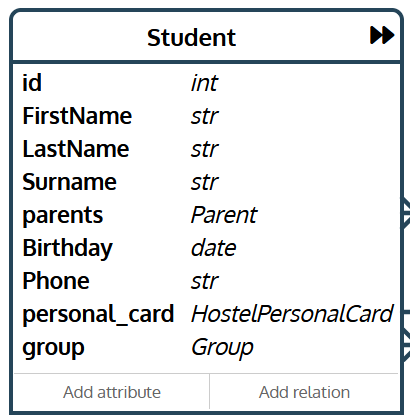


Рис. 3. Таблиця Student

Ця таблиця призначення для збереження інформації про самого студента. Тут зберігаються його айді, ПІБ, дата народження, номер телефону, інформація про батьків, групу та персональну карту. Зв’язок між студентом та персональною картою один до одного (фактично, це просто винос данних, які не завжди потрібно підтягувати). Також, зв’язок між студентом і групою один до багатьох (в одній групі багато студентів, в одного студента лише одна група). Таблиця полів:

| **Field Name** | **DataType** | **PK** | **FK** | **Description** |
| --- | --- | --- | --- | --- |
| id | int | + | - | Айді одного з батьків студента |
| FirstName | Str(32) | - | - | Ім’я |
| LastName | Str(32) | - | - | Прізвище |
| Surname | Str(32) | - | - | По батькові, не обов’язкове |
| Birthday | Date | - | - | Дата народження студента |
| Phone | Str(12) | - | - | Номер телефону. Максимальна довжина 12 символів |
| Personal\_card | int | - | + | Айді персональної карти студента |
| Group | str | - | + | Назва групи, до якої відноситься студент |

Таблиця Group

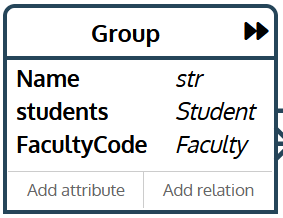


Рис. 4. Таблиця Group

У цій таблиці зберігається інформація про групи в університеті. У ній є назва групи, яка повинна бути унікальною і є PK. Також група посилається на факультет, до якого належить. Зв’язок один до багатьох (один факультет, багато груп). Таблиця полів:

| **Field Name** | **DataType** | **PK** | **FK** | **Description** |
| --- | --- | --- | --- | --- |
| Name | str | + | - | Унікальна назва групи (шифр) |
| FacultyCode | int | - | + | Посилання на факультет, до якого відноситься група |

Таблиця Faculty

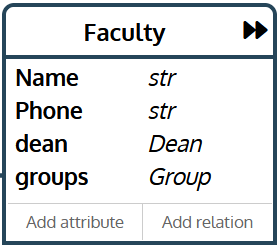


Рис. 5. Таблиця Faculty

Таблиця має інформацію про телефон факультету, його назву, посилання на айді деканата. Зв’язок між деканатом і деканом один до одного (фактично просто винесення інформації про декана в окрему таблицю для оптимізації запитів). Таблиця полів:

| **Field Name** | **DataType** | **PK** | **FK** | **Description** |
| --- | --- | --- | --- | --- |
| Name | str | + | - | Назва факультету |
| Phone | Str(12) | - | - | Номер факультету |
| dean | int | - | + | Айді деканата факультету |

Таблиця Dean

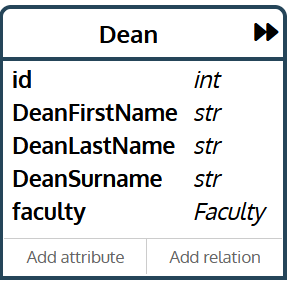


Рис. 6. Таблиця Dean

У цій таблиці зберігається інформація про декатів різних факультетів. Один декан може бути лише в одного факультету. Є унікальний айді та ПІБ. Таблиця полів:

| **Field Name** | **DataType** | **PK** | **FK** | **Description** |
| --- | --- | --- | --- | --- |
| id | int | + | - | Айді декана |
| DeanFirstName | Str(32) | - | - | Прізвище |
| DeanLastName | Str(32) | - | - | Ім’я |
| DeanSurname | Str(32) | - | - | По батькові, не обов’язкове |

Таблиця HostelPersonalCard

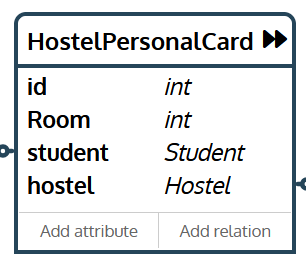


Рис. 7. Таблиця HostelPersonalCard

Ця таблиця зберігає персональні картки студентів, що проживають в гуртожитку. Вона має свій власний унікальний айді, кімнату, у якій проживає студент, посилання на самого студента (в одній кімнаті може бути багато студентів) та посилання на сам гуртожиток (на випадок, якщо університет має декілька гуртожитків).

| **Field Name** | **DataType** | **PK** | **FK** | **Description** |
| --- | --- | --- | --- | --- |
| id | int | + | - | Айді картки |
| Room | int(8) | - | - | Номер кімнати |
| Student | int | - | + | Посилання на студента картки |
| Hostel | int | - | + | Посилання на гуртожиток, в якому проживає студент |

Таблиця Hostel

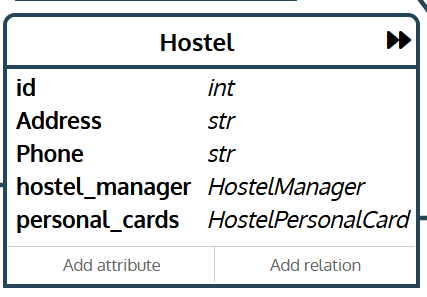


Рис. 8. Таблиця Hostel

Тут зберігається інформація про гуртожитки. Є айді самого гуртожитка, його адреса, номер телефону, посилання на коментанта (один до одного) та посилання на персональні картки студентів, що проживають в гуртожитку (один гуртожиток, багато студентів).

| **Field Name** | **DataType** | **PK** | **FK** | **Description** |
| --- | --- | --- | --- | --- |
| id | int | + | - | Айді гуртожитка |
| Address | Str(64) |  |  | Адреса гуртожитка |
| Hostel\_manager | id |  |  | Посилання на коменданта гуртожитка |
| Phone | Str(12) |  |  | Номер телефону гуртожитка |

Таблиця Hostel\_Manager

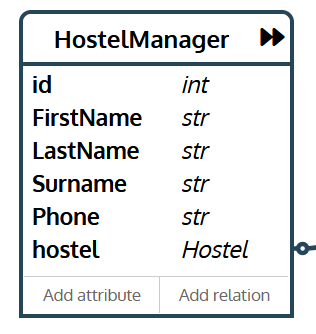


Рис. 9. Таблиця HostelManager

Ця таблиця зберігає інформацію про коменданта гуртожитка. Є персональні, контактні дані та посилання на гуртожиток, до якого комендант відноситься

| **Field Name** | **DataType** | **PK** | **FK** | **Description** |
| --- | --- | --- | --- | --- |
| id | int | + | - | Айді декана |
| FirstName | Str(32) | - | - | Прізвище |
| LastName | Str(32) | - | - | Ім’я |
| Surname | Str(32) | - | - | По батькові, не обов’язкове |
| Phone | Str(12) | - | - | Номер телефону |
| hostel | int | - | + | Посилання на гуртожиток |

SQL запит для створення структури БД:  
CREATE TABLE `faculty` (

`name` VARCHAR(64) PRIMARY KEY,

`phone` VARCHAR(16) NOT NULL

);

CREATE TABLE `dean` (

`id` TINYINT PRIMARY KEY AUTO\_INCREMENT,

`deanfirstname` VARCHAR(255) NOT NULL,

`deanlastname` VARCHAR(32) NOT NULL,

`deansurname` VARCHAR(32) NOT NULL,

`faculty` VARCHAR(64)

);

CREATE INDEX `idx\_dean\_\_faculty` ON `dean` (`faculty`);

ALTER TABLE `dean` ADD CONSTRAINT `fk\_dean\_\_faculty` FOREIGN KEY (`faculty`) REFERENCES `faculty` (`name`) ON DELETE SET NULL;

CREATE TABLE `group` (

`name` VARCHAR(16) PRIMARY KEY,

`facultycode` VARCHAR(64) NOT NULL

);

CREATE INDEX `idx\_group\_\_facultycode` ON `group` (`facultycode`);

ALTER TABLE `group` ADD CONSTRAINT `fk\_group\_\_facultycode` FOREIGN KEY (`facultycode`) REFERENCES `faculty` (`name`) ON DELETE CASCADE;

CREATE TABLE `hostel` (

`id` TINYINT PRIMARY KEY AUTO\_INCREMENT,

`address` VARCHAR(64) NOT NULL,

`phone` VARCHAR(12) NOT NULL

);

CREATE TABLE `hostelmanager` (

`id` TINYINT PRIMARY KEY AUTO\_INCREMENT,

`firstname` VARCHAR(32) NOT NULL,

`lastname` VARCHAR(32) NOT NULL,

`surname` VARCHAR(32) NOT NULL,

`phone` VARCHAR(12) NOT NULL,

`hostel` TINYINT NOT NULL

);

CREATE INDEX `idx\_hostelmanager\_\_hostel` ON `hostelmanager` (`hostel`);

ALTER TABLE `hostelmanager` ADD CONSTRAINT `fk\_hostelmanager\_\_hostel` FOREIGN KEY (`hostel`) REFERENCES `hostel` (`id`);

CREATE TABLE `parent` (

`id` INTEGER PRIMARY KEY AUTO\_INCREMENT,

`firstname` VARCHAR(32) NOT NULL,

`lastname` VARCHAR(32) NOT NULL,

`surname` VARCHAR(32) NOT NULL

);

CREATE TABLE `student` (

`id` TINYINT PRIMARY KEY AUTO\_INCREMENT,

`firstname` VARCHAR(32) NOT NULL,

`lastname` VARCHAR(32) NOT NULL,

`surname` VARCHAR(32) NOT NULL,

`birthday` DATE,

`phone` VARCHAR(12) NOT NULL,

`group` VARCHAR(16) NOT NULL

);

CREATE INDEX `idx\_student\_\_group` ON `student` (`group`);

ALTER TABLE `student` ADD CONSTRAINT `fk\_student\_\_group` FOREIGN KEY (`group`) REFERENCES `group` (`name`) ON DELETE CASCADE;

CREATE TABLE `hostelpersonalcard` (

`id` SMALLINT PRIMARY KEY AUTO\_INCREMENT,

`room` TINYINT NOT NULL,

`student` TINYINT NOT NULL,

`hostel` TINYINT NOT NULL

);

CREATE INDEX `idx\_hostelpersonalcard\_\_hostel` ON `hostelpersonalcard` (`hostel`);

CREATE INDEX `idx\_hostelpersonalcard\_\_student` ON `hostelpersonalcard` (`student`);

ALTER TABLE `hostelpersonalcard` ADD CONSTRAINT `fk\_hostelpersonalcard\_\_hostel` FOREIGN KEY (`hostel`) REFERENCES `hostel` (`id`);

ALTER TABLE `hostelpersonalcard` ADD CONSTRAINT `fk\_hostelpersonalcard\_\_student` FOREIGN KEY (`student`) REFERENCES `student` (`id`);

CREATE TABLE `parent\_student` (

`parent` INTEGER NOT NULL,

`student` TINYINT NOT NULL,

PRIMARY KEY (`parent`, `student`)

);

CREATE INDEX `idx\_parent\_student` ON `parent\_student` (`student`);

ALTER TABLE `parent\_student` ADD CONSTRAINT `fk\_parent\_student\_\_parent` FOREIGN KEY (`parent`) REFERENCES `parent` (`id`);

ALTER TABLE `parent\_student` ADD CONSTRAINT `fk\_parent\_student\_\_student` FOREIGN KEY (`student`) REFERENCES `student` (`id`)

Висновок: набуто практичних навичок щодо побудови концептуальних моделей баз даних із використанням діаграмних технологій (модель сутність-зв‘язок).