Санкт-Петербургский Национальный Исследовательский Университет ИТМО

Факультет Программной Инженерии и Компьютерной Техники



Вариант **№10117**

Лабораторная работа **№3**

По дисциплине

**Базы Данных**

*Выполнил*:

Студент группы P3110

Чжун Цзяцзюнь

*Преподаватель*:

Гаврилов Антон Валерьевич

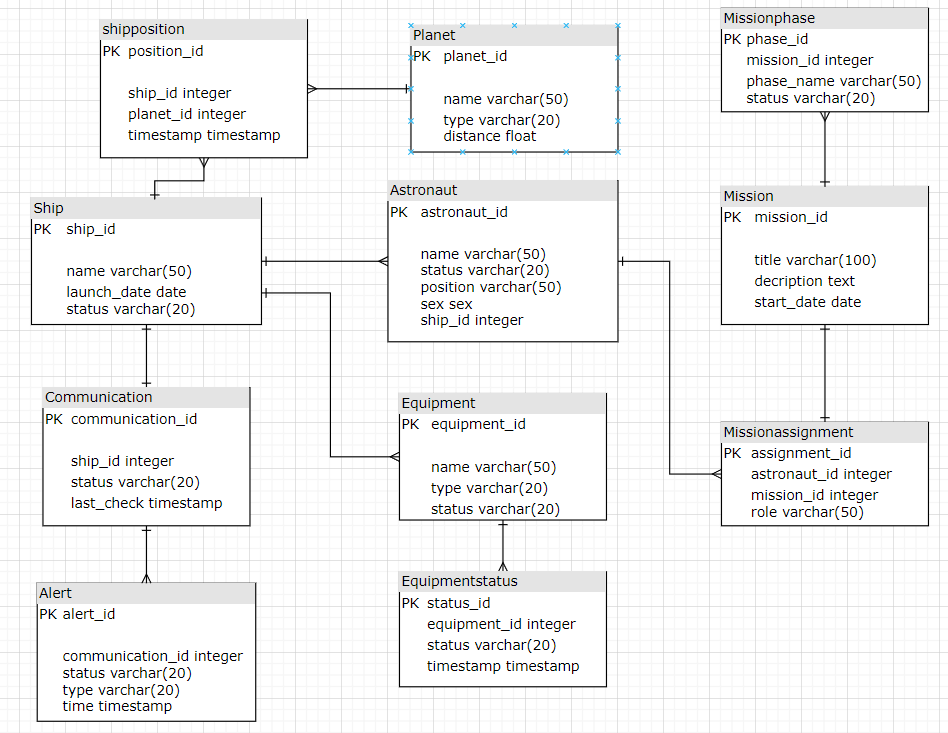
Санкт-Петербург 2025 г.

1. **Текст задания**

Для отношений, полученных при построении предметной области из лабораторной работы №1, выполните следующие действия:

* опишите функциональные зависимости для отношений полученной схемы (минимальное множество);
* приведите отношения в 3NF (как минимум). Постройте схему на основе 3NF (как минимум). Постройте схему на основе полученных отношений;
* опишите изменения в функциональных зависимостях, произошедшие после преобразования в 3NF (как минимум). Постройте схему на основе 3NF;
* преобразуйте отношения в BCNF. Докажите, что полученные отношения представлены в BCNF; Если ваша схема находится уже в BCNF, докажите это.
* Какие денормализации будут полезны для вашей схемы? Приведите подробное описание;

Придумайте триггер и связанную с ним функцию, относящиеся к вашей предметной области, согласуйте их с преподавателем и реализуйте на языке PL/pgSQL.

****

1. **Функциональные зависимости**

|  |
| --- |
| Ship: id → (name, launch\_date, status)  Communication: id → (ship\_id, status, last\_check)  Alert: id → (communication\_id, status, type, time)  Ship\_position: id → (ship\_id, planet\_id, timestamp)  Planet: id → (name, type, distance)  Astronaut: id → (name, status, position, sex, ship\_id)  Equipment: id → (name, type, status)  Equipment\_status: id → (equipment\_id, status, timestamp)  Mission\_assignment: id → (astronaut\_id, mission\_id, role)  Mission: id → (title, description, start\_date)  Mission\_phase: id → (mission\_id, phase\_name, status) |

1. **Нормальные формы**

|  |
| --- |
| 1NF: Модель удовлетворяет 1NF, так как все атрибуты атомарны, и отсутствуют повторяющиеся группы.  2NF: Отношение находится во 2NF, если оно находится в 1NF и все его неключевые атрибуты полностью функционально зависят от первичного ключа. (например, Ship: id → name, launch\_date, status) Моя модель удовлетворяет 2NF, так как все неключевые атрибуты полностью функционально зависят от первичных ключей.  3NF: Отношение находится в 3NF, если оно находится во 2NF и не содержит транзитивных зависимостей. Моя модель удовлетворяет 3NF, так как нет транзитивных зависимостей. Все атрибуты зависят только от id. Все неключевые атрибуты зависят только от первичных ключей. |

1. **BCNF**

|  |
| --- |
| Отношение находится в BCNF,  если для каждой функциональной зависимости X →Y,  X является суперключом(первичный). Моя модель удовлетворяет BCNF, так как  для всех функциональных зависимостей X является суперключом. |

1. **Денормализация**

|  |
| --- |
| **Объединение связанных таблиц**:  В некоторых случаях, объединение таблиц может уменьшить количество операций JOIN и ускорить обработку запросов. Например, объединить Ship и Ship\_position, если часто требуется информация о корабле и его позиции.  **Добавление избыточных атрибутов**:  В некоторых случаях добавление избыточных атрибутов может улучшить производительность запросов. Например, в таблицу Mission добавить current\_phase\_status, чтобы избежать JOIN с Mission\_phase. |

1. **Функция на языке PL/pgSQL**

Функция на языке PL/pgSQL для автоматического обновления статуса миссии.

|  |
| --- |
| -- Функция на языке PL/pgSQL для автоматического обновление статуса миссии.  -- Связь между таблицами:  -- MISSION - mission\_id (FK) - MISSIONPHASE - phase\_id (FK)  -- Статусы: active, completed, planned.  CREATE OR REPLACE FUNCTION update\_mission\_status()  RETURNS TRIGGER AS $$  DECLARE      total\_phases INT; -- Общее количество этапов текущей задачи      completed\_phases INT; -- Количество завершенных этапов текущей задачи  BEGIN      -- Получить общее количество этапов текущей задачи      SELECT COUNT(\*) INTO total\_phases      FROM MISSIONPHASE      WHERE MISSION\_ID = NEW.MISSION\_ID;  -- Получение количества завершенных этапов текущей задачи  SELECT COUNT(\*) INTO completed\_phases      FROM MISSIONPHASE      WHERE MISSION\_ID = NEW.MISSION\_ID      AND STATUS = 'completed';      -- Если все этапы завершены, обновим статус задачи до “completed”      IF total\_phases = completed\_phases AND total\_phases > 0 THEN          UPDATE MISSION          SET STATUS = 'completed'          WHERE MISSION\_ID = NEW.MISSION\_ID;      END IF;      RETURN NEW;  END;  $$ LANGUAGE plpgsql;  CREATE TRIGGER mission\_phase\_status\_trigger  AFTER UPDATE OF STATUS ON MISSIONPHASE  FOR EACH ROW  WHEN (NEW.STATUS = 'completed')  -- Срабатывает только при обновлении статуса до "completed".  EXECUTE FUNCTION update\_mission\_status(); |

1. **Вывод**

При выполнении лабораторной работы я познакомился с понятием нормализации и денормализации. Научился определять функциональные зависимости модели, а также анализировать последнюю на соответствие различным нормальным формам. Познакомился с процедурным языком PL/pgSQL. Изучил эффективные способы денормализации схемы базы данных и ситуации, в которых возможно их применение.