

Чеклист по БД

Чеклист основан на заданиях рубежек прошлых лет

ЗАДАНИЕ МОЖЕТ НЕ СОВПАДАТЬ

Чеклист:

- ☐ 1) Ты написал инфологическую модель
 - ☐ Сущности готовы
 - ☐ 1 (/ 7 атт.) ☐ 4 (/ 7 атт.) ☐ 7 (/ 7 атт.)
 - ☐ 2 (/ 7 атт.) ☐ 5 (/ 7 атт.) ☐ 8 (/ 7 атт.)
 - ☐ 3 (/ 7 атт.) ☐ 6 (/ 7 атт.) ☐ 9 (/ 7 атт.)
 - ☐ Сущности имеют связи для дальнейших SQL запросов
 - ☐ Есть как минимум одна связь вида многие-ко-многим
- ☐ 2) Ты построил даталогическую модель
 - ☐ Перенесены все сущности
 - ☐ Корректно выставлены связи и ключи РК/FK
 - ☐ Корректно написаны соединения 1 to many и т.п.
 - ☐ Есть ограничения целостности
- ☐ 3) Ты написал SQL код на DDL
 - ☐ Каждая сущность как таблица создаётся через CREATE TABLE
 - ☐ В каждой из таблиц точно указаны все ключи
 - ☐ Ты не забыл про CHECK
- ☐ 4) Ты написал запрос в свою БД
 - ☐ Ты использовал оператор ANY в нём.
 - ☐ Ты объяснил запрос
- ☐ Рубежка сдана



Чеклист по БД

Чеклист основан на заданиях рубежек прошлых лет

ЗАДАНИЕ МОЖЕТ НЕ СОВПАДАТЬ

Чеклист:

- ☐ 1) Ты написал инфологическую модель
 - ☐ Сущности готовы
 - ☐ 1 (/ 7 атт.) ☐ 4 (/ 7 атт.) ☐ 7 (/ 7 атт.)
 - ☐ 2 (/ 7 атт.) ☐ 5 (/ 7 атт.) ☐ 8 (/ 7 атт.)
 - ☐ 3 (/ 7 атт.) ☐ 6 (/ 7 атт.) ☐ 9 (/ 7 атт.)
 - ☐ Сущности имеют связи для дальнейших SQL запросов
 - ☐ Есть как минимум одна связь вида многие-ко-многим
- ☐ 2) Ты построил даталогическую модель
 - ☐ Перенесены все сущности
 - ☐ Корректно выставлены связи и ключи РК/FK
 - ☐ Корректно написаны соединения 1 to many и т.п.
 - ☐ Есть ограничения целостности
- ☐ 3) Ты написал SQL код на DDL
 - ☐ Каждая сущность как таблица создаётся через CREATE TABLE
 - ☐ В каждой из таблиц точно указаны все ключи
 - ☐ Ты не забыл про CHECK
- ☐ 4) Ты написал запрос в свою БД
 - ☐ Ты использовал оператор ANY в нём.
 - ☐ Ты объяснил запрос
- ☐ Рубежка сдана



Чеклист по БД

Чеклист основан на заданиях рубежек прошлых лет

ЗАДАНИЕ МОЖЕТ НЕ СОВПАДАТЬ

Чеклист:

- ☐ 1) Ты написал инфологическую модель
 - ☐ Сущности готовы
 - ☐ 1 (/ 7 атт.) ☐ 4 (/ 7 атт.) ☐ 7 (/ 7 атт.)
 - ☐ 2 (/ 7 атт.) ☐ 5 (/ 7 атт.) ☐ 8 (/ 7 атт.)
 - ☐ 3 (/ 7 атт.) ☐ 6 (/ 7 атт.) ☐ 9 (/ 7 атт.)
 - ☐ Сущности имеют связи для дальнейших SQL запросов
 - ☐ Есть как минимум одна связь вида многие-ко-многим
- ☐ 2) Ты построил даталогическую модель
 - ☐ Перенесены все сущности
 - ☐ Корректно выставлены связи и ключи РК/FK
 - ☐ Корректно написаны соединения 1 to many и т.п.
 - ☐ Есть ограничения целостности
- ☐ 3) Ты написал SQL код на DDL
 - ☐ Каждая сущность как таблица создаётся через CREATE TABLE
 - ☐ В каждой из таблиц точно указаны все ключи
 - ☐ Ты не забыл про CHECK
- ☐ 4) Ты написал запрос в свою БД
 - ☐ Ты использовал оператор ANY в нём.
 - ☐ Ты объяснил запрос
- ☐ Рубежка сдана



1. Инфологическая Модель (ER)

Концепция данных:

- Сущности** (объекты, напр., Студент)
- Атрибуты** (свойства, напр., Имя)
- Связи** (отношения 1:1, 1:M, M:M)

2. Даталогическая Модель (Реляционная)

Схема таблиц (отношений) для СУБД.

- ER -> Таблицы:
 - Сущность -> Таблица
 - Атрибут -> Столбец (с типом данных)
 - 1:M -> FK (внешний ключ) на стороне 'M'.
 - M:M -> Отдельная **связующая таблица** (с двумя FK).
- Ключи:
 - PK** (Primary Key): Уникальный ID строки, не NULL.
 - FK** (Foreign Key): Ссылка на PK другой таблицы, обеспечивает связь.
- Целостность: PK (не NULL), Ссылочная (FK -> PK), Пользовательская (CHECK, UNIQUE).

3. SQL DDL (Создание Структуры)

Команды для создания объектов БД.

```
CREATE TABLE students (  
    st_id          INTEGER NOT NULL,  
    st_name        TEXT,  
    failed_courses INTEGER CHECK (failed_courses >= 0)  
);
```

- Типы (примеры): INTEGER, SERIAL (авто ID), TEXT, DATE, BOOLEAN.
- Ограничения: PRIMARY KEY, FOREIGN KEY REFERENCES ..., UNIQUE, NOT NULL, CHECK (условие).

4. SQL Запрос (с ANY)

Выборка/изменение данных (DML). SELECT ... FROM ... WHERE ...

- Оператор ANY:
 - Синтаксис: выражение оператор_сравнения ANY (подзапрос | массив)
 - Значение: TRUE, если сравнение верно **хотя бы для одного** значения из подзапроса/массива.
 - Пример: grade > ANY (ARRAY[2, 3]) (Оценка больше 2 **или** больше 3, т.е. > 2)
 - Пример: id = ANY (SELECT user_id FROM banned_users) (ID есть в списке забаненных)

1. Инфологическая Модель (ER)

Концепция данных:

- Сущности** (объекты, напр., Студент)
- Атрибуты** (свойства, напр., Имя)
- Связи** (отношения 1:1, 1:M, M:M)

2. Даталогическая Модель (Реляционная)

Схема таблиц (отношений) для СУБД.

- ER -> Таблицы:
 - Сущность -> Таблица
 - Атрибут -> Столбец (с типом данных)
 - 1:M -> FK (внешний ключ) на стороне 'M'.
 - M:M -> Отдельная **связующая таблица** (с двумя FK).
- Ключи:
 - PK** (Primary Key): Уникальный ID строки, не NULL.
 - FK** (Foreign Key): Ссылка на PK другой таблицы, обеспечивает связь.
- Целостность: PK (не NULL), Ссылочная (FK -> PK), Пользовательская (CHECK, UNIQUE).

3. SQL DDL (Создание Структуры)

Команды для создания объектов БД.

```
CREATE TABLE students (  
    st_id          INTEGER NOT NULL,  
    st_name        TEXT,  
    failed_courses INTEGER CHECK (failed_courses >= 0)  
);
```

- Типы (примеры): INTEGER, SERIAL (авто ID), TEXT, DATE, BOOLEAN.
- Ограничения: PRIMARY KEY, FOREIGN KEY REFERENCES ..., UNIQUE, NOT NULL, CHECK (условие).

4. SQL Запрос (с ANY)

Выборка/изменение данных (DML). SELECT ... FROM ... WHERE ...

- Оператор ANY:
 - Синтаксис: выражение оператор_сравнения ANY (подзапрос | массив)
 - Значение: TRUE, если сравнение верно **хотя бы для одного** значения из подзапроса/массива.
 - Пример: grade > ANY (ARRAY[2, 3]) (Оценка больше 2 **или** больше 3, т.е. > 2)
 - Пример: id = ANY (SELECT user_id FROM banned_users) (ID есть в списке забаненных)

1. Инфологическая Модель (ER)

Концепция данных:

- Сущности** (объекты, напр., Студент)
- Атрибуты** (свойства, напр., Имя)
- Связи** (отношения 1:1, 1:M, M:M)

2. Даталогическая Модель (Реляционная)

Схема таблиц (отношений) для СУБД.

- ER -> Таблицы:
 - Сущность -> Таблица
 - Атрибут -> Столбец (с типом данных)
 - 1:M -> FK (внешний ключ) на стороне 'M'.
 - M:M -> Отдельная **связующая таблица** (с двумя FK).
- Ключи:
 - PK** (Primary Key): Уникальный ID строки, не NULL.
 - FK** (Foreign Key): Ссылка на PK другой таблицы, обеспечивает связь.
- Целостность: PK (не NULL), Ссылочная (FK -> PK), Пользовательская (CHECK, UNIQUE).

3. SQL DDL (Создание Структуры)

Команды для создания объектов БД.

```
CREATE TABLE students (  
    st_id          INTEGER NOT NULL,  
    st_name        TEXT,  
    failed_courses INTEGER CHECK (failed_courses >= 0)  
);
```

- Типы (примеры): INTEGER, SERIAL (авто ID), TEXT, DATE, BOOLEAN.
- Ограничения: PRIMARY KEY, FOREIGN KEY REFERENCES ..., UNIQUE, NOT NULL, CHECK (условие).

4. SQL Запрос (с ANY)

Выборка/изменение данных (DML). SELECT ... FROM ... WHERE ...

- Оператор ANY:
 - Синтаксис: выражение оператор_сравнения ANY (подзапрос | массив)
 - Значение: TRUE, если сравнение верно **хотя бы для одного** значения из подзапроса/массива.
 - Пример: grade > ANY (ARRAY[2, 3]) (Оценка больше 2 **или** больше 3, т.е. > 2)
 - Пример: id = ANY (SELECT user_id FROM banned_users) (ID есть в списке забаненных)