ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ им. проф. М.А. БОНЧ-БРУЕВИЧА» (СПбГУТ)

Факультет Информационных технологий и программной инженерии
Кафедра Систем обработки данных

ОТЧЕТ

по практическому занятию №1

Тема: «Резервное копирование и восстановление»

Выполнила: студентка 3 курса, гр. ИБ-31вп
_____ Киселева А. Е.

Принял:
_____ Медведев С. А.

Санкт-Петербург, 2025 г.

Содержание

Отчет по резервному копированию

Описание предметной области

База данных описывает результаты полевых опытов сельскохозяйственного института, который изучает влияние предшественников и режимов ухода за почвой на урожайность зерновых культур в Новосибирской области. Для каждой культуры фиксируются поле, год, агротехнические условия и урожай, что позволяет анализировать эффективность технологий и подбирать оптимальные схемы севооборота. Система служит основой для отчетности, прогнозов и планирования ресурсов.

Схема базы данных

```sql
CREATE TABLE cultures (
    id SERIAL PRIMARY KEY,
    name TEXT UNIQUE
);

CREATE TABLE predecessors (
    id SERIAL PRIMARY KEY,
    name TEXT UNIQUE
);

CREATE TABLE conditions (
    id SERIAL PRIMARY KEY,
    name TEXT UNIQUE
);

CREATE TABLE experiments (
    id SERIAL PRIMARY KEY,
    year INT,
    field INT,
    culture_id INT REFERENCES cultures(id),
    predecessor_id INT REFERENCES predecessors(id),
    condition_id INT REFERENCES conditions(id),
    yield NUMERIC(5,2)
);
```

Примеры запросов на вставку данных

```sql
INSERT INTO cultures (name) VALUES ('Горох'), ('Овёс'), ('Пшеница'), ('Ячмень');
INSERT INTO predecessors (name) VALUES ('Овёс'), ('Пар'), ('Пшеница'), ('Ячмень');
INSERT INTO conditions (name) VALUES ('Без травы'), ('С травой');
INSERT INTO experiments (year, field, culture_id, predecessor_id, condition_id, yield)
VALUES (2015, 1, 1, 1, 2, 19.10),
       (2015, 1, 4, 2, 1, 31.80),
       (2015, 2, 4, 1, 2, 22.00),
```

```
       (2015, 2, 1, 4, 2, 16.00);
```

Ход экспериментов по резервному копированию

Последовательный запуск `make run_all` выполняет сборку контейнера, все варианты резервного копирования и восстановление данных. Ниже приведен полный вывод, начиная с шага «Делаем резервное копирование данных в формате custom».

```
/Applications/Xcode.app/Contents/Developer/usr/bin/make build
docker compose down -v && docker compose up -d --build
[+] Running 3/3
 ✔ Container postgres-db1    Removed
  ↪  0.1s
 ✔ Volume 1_backup_pgdata1   Removed
  ↪  0.0s
 ✔ Network 1_backup_default  Removed
  ↪  0.2s
[+] Building 2.9s (10/10) FINISHED
 => [internal] load local bake definitions
  ↪  0.0s
 => => reading from stdin 525B
  ↪  0.0s
 => [internal] load build definition from Dockerfile
  ↪  0.0s
 => => transferring dockerfile: 188B
  ↪  0.0s
 => [internal] load metadata for docker.io/library/postgres:18.0-alpine
  ↪  2.7s
 => [auth] library/postgres:pull token for registry-1.docker.io
  ↪  0.0s
 => [internal] load .dockerignore
  ↪  0.0s
 => => transferring context: 2B
  ↪  0.0s
 => [internal] load build context
  ↪  0.0s
 => => transferring context: 174B
  ↪  0.0s
 => [1/2] FROM docker.io/library/postgres:18.0-alpine@sha256:48c8ad3a7284b82be4482a
  ↪  52076d47d879fd6fb084a1cbfccbd551f9331b0e40               0.0s
 => => resolve docker.io/library/postgres:18.0-alpine@sha256:48c8ad3a7284b82be4482a
  ↪  52076d47d879fd6fb084a1cbfccbd551f9331b0e40               0.0s
 => CACHED [2/2] COPY init/ /docker-entrypoint-initdb.d/
  ↪  0.0s
 => exporting to image
  ↪  0.0s
```

```
 => => exporting layers
  ↪  0.0s
 => => exporting manifest
  ↪  sha256:14aa4934d3a2289ffac6ced2ff600abd3b425f643af013b0a9cd55cb858a7d7a
  ↪  0.0s
 => => exporting config
  ↪  sha256:5be411ff2c7ab27bbce4e5dce2d69a6a94aa0a68b9c608140aabe8e70289c23c
  ↪  0.0s
 => => exporting attestation manifest
  ↪  sha256:825ba994a161a134726b838f251ce44c4f76be35c05c8b3b550bd464a4366b1a
  ↪  0.0s
 => => exporting manifest list
  ↪  sha256:38b78d4987957f41ffff507e62d22f0f75ed52de6932ee7232e05a3a0b2fbb54
  ↪  0.0s
 => => naming to docker.io/library/1_backup-db1:latest
  ↪  0.0s
 => => unpacking to docker.io/library/1_backup-db1:latest
  ↪  0.0s
 => resolving provenance for metadata file
  ↪  0.0s
[+] Running 4/4
 ✔ 1_backup-db1              Built
  ↪  0.0s
 ✔ Network 1_backup_default  Created
  ↪  0.0s
 ✔ Volume 1_backup_pgdata1   Created
  ↪  0.0s
 ✔ Container postgres-db1    Started
  ↪  0.1s
```

Делаем резервное копирование данных в формате custom.

```
/Applications/Xcode.app/Contents/Developer/usr/bin/make 1_backup_custom
./scripts/backup_custom.sh db1_custom.dump
Waiting for db1 to become ready...
Writing custom-format backup to
 ↪  /Users/ami/source/manipulate/1_BACKUP/backups/db1_custom.dump
Backup completed.
```

Проверяем какие данные содержатся в данный момент и выполняем обновление.

```
/Applications/Xcode.app/Contents/Developer/usr/bin/make 7_show_update_experiment
./scripts/show_update_experiment.sh
Waiting for db1 to become ready...
BEGIN
 id | year | field | culture_id | predecessor_id | condition_id | yield
----+------+-------+------------+----------------+--------------+-------
```

```
 14 | 2015 |     7 |          3 |              3 |            1 | 28.30
(1 row)

UPDATE 1
 id | year | field | culture_id | predecessor_id | condition_id | yield
----+------+-------+------------+----------------+--------------+-------
 14 | 2025 |     7 |          3 |              3 |            1 | 28.30
(1 row)

Выполняем восстановление базы данных и проверяем, что изменения откатились.

COMMIT
/Applications/Xcode.app/Contents/Developer/usr/bin/make 2_restore_custom
./scripts/restore_custom.sh ./backups/db1_custom.dump
Waiting for db1 to become ready...
Restoring db1 from ./backups/db1_custom.dump
Restore completed.
/Applications/Xcode.app/Contents/Developer/usr/bin/make 8_show_experiment
./scripts/show_experiment.sh
Waiting for db1 to become ready...
 id | year | field | culture_id | predecessor_id | condition_id | yield
----+------+-------+------------+----------------+--------------+-------
 14 | 2015 |     7 |          3 |              3 |            1 | 28.30
(1 row)

Выполняем plain-бэкапы с разными вариантами параметра `Sections` (all, pre, data,
 ↪  post).

/Applications/Xcode.app/Contents/Developer/usr/bin/make 3_backup_sections
./scripts/backup_plain_sections.sh
Waiting for db1 to become ready...
Dumping pre-data section
Dumping data section
Dumping post-data section
Section-specific backups are stored in
 ↪  /Users/ami/source/manipulate/1_BACKUP/backups/plain_sections
/Applications/Xcode.app/Contents/Developer/usr/bin/make 3_sections_pre
./scripts/backup_plain_sections.sh pre-data
Waiting for db1 to become ready...
Dumping pre-data section
Section-specific backups are stored in
 ↪  /Users/ami/source/manipulate/1_BACKUP/backups/plain_sections
/Applications/Xcode.app/Contents/Developer/usr/bin/make 3_sections_data
./scripts/backup_plain_sections.sh data
Waiting for db1 to become ready...
Dumping data section
Section-specific backups are stored in
 ↪  /Users/ami/source/manipulate/1_BACKUP/backups/plain_sections
```

```
/Applications/Xcode.app/Contents/Developer/usr/bin/make 3_sections_post
./scripts/backup_plain_sections.sh post-data
Waiting for db1 to become ready...
Dumping post-data section
Section-specific backups are stored in
 ↪  /Users/ami/source/manipulate/1_BACKUP/backups/plain_sections
/Applications/Xcode.app/Contents/Developer/usr/bin/make 4_backup_types
```

Проверяем влияние настроек `Type of objects` при режимах Only Schema и Only Data.

```
./scripts/backup_plain_types.sh
Waiting for db1 to become ready...
Dumping schema-only script
Dumping data-only script
Type-specific backups are stored in
 ↪  /Users/ami/source/manipulate/1_BACKUP/backups/plain_types
/Applications/Xcode.app/Contents/Developer/usr/bin/make 4_type_schema
./scripts/backup_plain_types.sh schema
Waiting for db1 to become ready...
Dumping schema-only script
Type-specific backups are stored in
 ↪  /Users/ami/source/manipulate/1_BACKUP/backups/plain_types
/Applications/Xcode.app/Contents/Developer/usr/bin/make 4_type_data
./scripts/backup_plain_types.sh data
Waiting for db1 to become ready...
Dumping data-only script
Type-specific backups are stored in
 ↪  /Users/ami/source/manipulate/1_BACKUP/backups/plain_types
```

Создаем резервную копию в формате `Directory` и выполняем восстановление.

```
/Applications/Xcode.app/Contents/Developer/usr/bin/make 5_backup_directory
./scripts/backup_directory.sh
Waiting for db1 to become ready...
Creating directory-format backup at /Users/ami/source/manipulate/1_BACKUP/backups/d⌋
 ↪  irectory/db1_directory_20251103_194226
[+] Copying 1/1
 ✔ postgres-db1 copy postgres-db1:/tmp/db1_directory_20251103_194226 to /Users/ami/⌋
  ↪  source/manipulate/1_BACKUP/backups/directory/db1_directory_20251103_194226
  ↪  Copied0.0s
Directory backup stored at /Users/ami/source/manipulate/1_BACKUP/backups/directory/⌋
 ↪  db1_directory_20251103_194226
/Applications/Xcode.app/Contents/Developer/usr/bin/make 6_restore_directory
./scripts/restore_directory.sh
Using latest directory backup: /Users/ami/source/manipulate/1_BACKUP/backups/direct⌋
 ↪  ory/db1_directory_20251103_194226
Waiting for db1 to become ready...
```

```
Copying /Users/ami/source/manipulate/1_BACKUP/backups/directory/db1_directory_20251
 ↪  103_194226 into container path /tmp/db1_directory_20251103_194226
[+] Copying 1/1
 ✔ postgres-db1 copy /Users/ami/source/manipulate/1_BACKUP/backups/directory/db1_di
  ↪  rectory_20251103_194226 to postgres-db1:/tmp/db1_directory_20251103_194226
  ↪  Copied0.0s
Restoring db1 from directory backup
Directory restore completed.
```