Migration Rollbacks

CREATE TABLE migration

Forward migration

```
CREATE TABLE customers (
    id SERIAL PRIMARY KEY,
    name TEXT NOT NULL
);
```

Rollback migration

```
DROP TABLE customers;
```

ADD COLUMN migration

Forward migration

```
ALTER TABLE customers
    ADD COLUMN date_of_birth TIMESTAMP;
```

Rollback migration

```
ALTER TABLE customers
    DROP COLUMN date_of_birth;
```

SET NOT NULL migration

Forward migration

```
ALTER TABLE customers
    ALTER COLUMN date_of_birth SET NOT NULL;
```

Rollback migration

```
ALTER TABLE customers
    ALTER COLUMN date_of_birth DROP NOT NULL;
```

SET DEFAULT migration

Forward migration

```
ALTER TABLE customers
    ALTER COLUMN date_of_birth SET DEFAULT now();
```

Rollback migration

```
ALTER TABLE customers
    ALTER COLUMN date_of_birth DROP DEFAULT;
```

RENAME COLUMN migration

Forward migration

```
ALTER TABLE customers
    RENAME COLUMN date_of_birth TO dob;
```

Rollback migration

```
ALTER TABLE customers
    RENAME COLUMN dob TO date_of_birth;
```

CREATE TABLE migration

Forward migration

```
CREATE TABLE orders (
    id SERIAL PRIMARY KEY,
    dollar_amount_spent NUMERIC,
    customer_id INT NOT NULL,
    CONSTRAINT fk_orders_customers
        FOREIGN KEY(customer_id)
        REFERENCES customers(id)
        ON DELETE CASCADE
);
```

Rollback migration

```
DROP TABLE orders;
```

DROP CONSTRAINT migration

Forward migration

```
ALTER TABLE orders
    DROP CONSTRAINT fk_orders_customers;
```

Rollback migration

```
ALTER TABLE orders
ADD CONSTRAINT fk_orders_customers
FOREIGN KEY (customer_id)
REFERENCES customers(id)
ON DELETE CASCADE;
```

Multi-statement migration

Forward migration

```
CREATE TABLE table1 (id SERIAL PRIMARY KEY);
CREATE TABLE table2 (id SERIAL PRIMARY KEY);
CREATE TABLE table3 (id SERIAL PRIMARY KEY);
```

Rollback migration

```
DROP TABLE table3;
DROP TABLE table2;
DROP TABLE table1;
```

Recap

Too much syntax to memorize

Refer to documentation as needed and remember these key points:

Schema migrations manipulate structure database tables, not data values

Each migration introduces minimal change

Each migration is non-breaking, maintains valid database

Rollback queries undo migrations

Multi-statement rollbacks execute statements in reverse order of forward migration