

Nannestad Pizzabar – Ordresøk (Node + Express + pg + PostgreSQL)

Mål: Lage en veldig enkel nettside der dere skriver inn **ordrenr (ordre_id)** og trykker på knapper for å hente:

- **Ordreinfo** (kun fra `ordre`)
- **Produktinfo** (kun fra `produkt`)
- **Kundeinfo** (kun fra `kunde`)
- **All info** (ordre + kunde + produkt)
- **Alle ordre** (kort liste for alle ordre) → **GET /alleordre**

Krav: Bruk **INNER JOIN** når dere trenger info fra flere tabeller.

Del 1: Lag database og tabeller i pgAdmin

1) Opprett database

1. Åpne **pgAdmin**
2. Høyreklikk **Databases** → **Create** → **Database...**
3. Name: `nannestad_pizzabar`
4. **Save**

2) Lag tabeller (Query Tool)

1. Klikk på databasen `nannestad_pizzabar`
2. **Tools** → **Query Tool**
3. Kjør:

```
CREATE TABLE kunde (
    kunde_id SERIAL PRIMARY KEY,
    navn TEXT NOT NULL,
    telefon TEXT,
    epost TEXT UNIQUE NOT NULL
);
```

```
CREATE TABLE produkt (
    produkt_id SERIAL PRIMARY KEY,
    navn TEXT NOT NULL,
    kategori TEXT NOT NULL,
    pris NUMERIC(10,2) NOT NULL
);
```

```
CREATE TABLE ordre (
    ordre_id SERIAL PRIMARY KEY,
    kunde_id INT NOT NULL,
    produkt_id INT NOT NULL,
    antall INT NOT NULL,
```

```
ordre_dato DATE NOT NULL,  
status TEXT NOT NULL,  
FOREIGN KEY (kunde_id) REFERENCES kunde (kunde_id),  
FOREIGN KEY (produkt_id) REFERENCES produkt (produkt_id)  
);
```

Del 2: Fyll databasen med testdata

Kjør dette i Query Tool:

```
INSERT INTO kunde (navn, telefon, epost) VALUES  
('Ola Nordmann', '90000001', 'ola@nannestad.no'),  
('Sara Hansen', '90000002', 'sara@nannestad.no'),  
('Ali Ahmad', '90000003', 'ali@nannestad.no'),  
('Kari Nilsen', '90000004', 'kari@nannestad.no'),  
('Emma Berg', '90000005', 'emma@nannestad.no');  
  
INSERT INTO produkt (navn, kategori, pris) VALUES  
('Margherita', 'Pizza', 159.00),  
('Pepperoni', 'Pizza', 189.00),  
('Kebabpizza', 'Pizza', 205.00),  
('Cola 0.5L', 'Drikke', 45.00),  
('Fanta 0.5L', 'Drikke', 45.00),  
('Hvitløksdipp', 'Tilbehør', 25.00);  
  
INSERT INTO ordre (kunde_id, produkt_id, antall, ordre_dato, status) VALUES  
(1, 2, 1, '2026-02-01', 'mottatt'),  
(2, 1, 2, '2026-02-01', 'klar'),  
(3, 3, 1, '2026-02-02', 'under arbeid'),  
(4, 4, 3, '2026-02-02', 'mottatt'),  
(5, 2, 2, '2026-02-03', 'klar'),  
(1, 6, 1, '2026-02-03', 'klar'),  
(2, 5, 2, '2026-02-04', 'mottatt'),  
(3, 1, 1, '2026-02-04', 'klar'),  
(4, 2, 1, '2026-02-05', 'under arbeid'),  
(5, 3, 1, '2026-02-05', 'mottatt'),  
(1, 4, 2, '2026-02-06', 'mottatt'),  
(2, 6, 3, '2026-02-06', 'klar');
```

Del 3: Oppsett av Node/Express i VS Code

1) Lag prosjektmappe og init

1. Lag mappe: **nannestad-pizzabar**
2. Åpne mappen i **VS Code**
3. Kjør i terminal:

```
npm init -y
```

2) Installer pakker (viktig!)

Dere må ha:

- **express**
- **pg**
- **ejs**
- **dotenv**

Kjør:

```
npm install express pg ejs dotenv
```

3) Filstruktur

Lag:

```
nannestad-pizzabar/
  server.js
  .env
  views/
    index.ejs
```

4) Bruk ES modules

I **package.json**, legg inn:

```
"type": "module"
```

5) Lag **.env**

Bytt ut **PASSORD_HER** med ditt postgres-passord:

```
PORT=3000
DB_HOST=localhost
DB_PORT=5432
DB_USER=postgres
DB_PASSWORD=PASSORD_HER
DB_NAME=nannestad_pizzabar
```

Del 4: Lag server.js

Kopier inn:

```
import express from "express";
import pg from "pg";
import dotenv from "dotenv";

dotenv.config();

const app = express();
app.use(express.urlencoded({ extended: true }));
app.set("view engine", "ejs");

const { Pool } = pg;

const pool = new Pool({
  host: process.env.DB_HOST,
  port: Number(process.env.DB_PORT),
  user: process.env.DB_USER,
  password: process.env.DB_PASSWORD,
  database: process.env.DB_NAME,
});

app.get("/", (req, res) => {
  res.render("index", { rows: null, title: "Klar", error: null });
});

// 5) GI ALLE ORDRE (GET) - kort liste: produktnavn, ordredato, kundenavn
app.get("/alleordre", async (req, res) => {
  try {
    const sql = `
      SELECT
        produkt.navn AS produktnavn,
        ordre.ordre_dato,
        kunde.navn AS kundenavn
      FROM ordre
      INNER JOIN kunde ON ordre.kunde_id = kunde.kunde_id
      INNER JOIN produkt ON ordre.produkt_id = produkt.produkt_id
      ORDER BY ordre.ordre_dato DESC, kundenavn ASC;
    `;
    const result = await pool.query(sql);
    res.render("index", { rows: result.rows, title: "Alle ordre (kort)", error: null });
  } catch (err) {
    console.error(err);
    res.render("index", { rows: null, title: "Alle ordre (kort)", error: "Databasefeil. Se terminal." });
  }
});

// 1) VIS ORDREINFO (kun ordre)
```

```
app.post("/ordreinfo", async (req, res) => {
  const ordreId = Number(req.body.ordre_id);
  if (!ordreId) return res.render("index", { rows: null, title: "Ordreinfo", error: "Skriv inn et gyldig ordrenr (tall)." });

  try {
    const sql = `
      SELECT ordre_id, kunde_id, produkt_id, antall, ordre_dato, status
      FROM ordre
      WHERE ordre_id = $1;
    `;
    const result = await pool.query(sql, [ordreId]);
    res.render("index", { rows: result.rows, title: "Ordreinfo", error: null });
  } catch (err) {
    console.error(err);
    res.render("index", { rows: null, title: "Ordreinfo", error: "Databasefeil. Se terminal." });
  }
});

// 2) VIS PRODUKTINFO (kun produkt) - finn produkt via ordre (INNER JOIN)
app.post("/produktinfo", async (req, res) => {
  const ordreId = Number(req.body.ordre_id);
  if (!ordreId) return res.render("index", { rows: null, title: "Produktinfo", error: "Skriv inn et gyldig ordrenr (tall)." });

  try {
    const sql = `
      SELECT produkt.produkt_id, produkt.navn, produkt.kategori, produkt.pris
      FROM ordre
      INNER JOIN produkt ON ordre.produkt_id = produkt.produkt_id
      WHERE ordre.ordre_id = $1;
    `;
    const result = await pool.query(sql, [ordreId]);
    res.render("index", { rows: result.rows, title: "Produktinfo", error: null });
  } catch (err) {
    console.error(err);
    res.render("index", { rows: null, title: "Produktinfo", error: "Databasefeil. Se terminal." });
  }
});

// 3) VIS KUNDEINFO (kun kunde) - finn kunde via ordre (INNER JOIN)
app.post("/kundeinfo", async (req, res) => {
  const ordreId = Number(req.body.ordre_id);
  if (!ordreId) return res.render("index", { rows: null, title: "Kundeinfo", error: "Skriv inn et gyldig ordrenr (tall)." });

  try {
    const sql = `
      SELECT kunde.kunde_id, kunde.navn, kunde.telefon, kunde.epost
      FROM ordre
      INNER JOIN kunde ON ordre.kunde_id = kunde.kunde_id
      WHERE ordre.ordre_id = $1;
    `;
  
```

```
`;
const result = await pool.query(sql, [ordreId]);
res.render("index", { rows: result.rows, title: "Kundeinfo", error: null });
} catch (err) {
  console.error(err);
  res.render("index", { rows: null, title: "Kundeinfo", error: "Databasefeil. Se terminal." });
}
});

// 4) VIS ALL INFO (ordre + kunde + produkt)
app.post("/allinfo", async (req, res) => {
  const ordreId = Number(req.body.ordre_id);
  if (!ordreId) return res.render("index", { rows: null, title: "All info", error: "Skriv inn et gyldig ordrenr (tall)." });

  try {
    const sql = `
      SELECT
        ordre.ordre_id,
        ordre.antall,
        ordre.ordre_dato,
        ordre.status,
        kunde.navn AS kundenavn,
        kunde.epost,
        produkt.navn AS produktnavn,
        produkt.kategori,
        produkt.pris
      FROM ordre
      INNER JOIN kunde ON ordre.kunde_id = kunde.kunde_id
      INNER JOIN produkt ON ordre.produkt_id = produkt.produkt_id
      WHERE ordre.ordre_id = $1;
    `;
    const result = await pool.query(sql, [ordreId]);
    res.render("index", { rows: result.rows, title: "All info", error: null });
  } catch (err) {
    console.error(err);
    res.render("index", { rows: null, title: "All info", error: "Databasefeil. Se terminal." });
  }
});

app.listen(process.env.PORT, () => {
  console.log(`Server kjører på http://localhost:${process.env.PORT}`);
});
```

Del 5: Lag nettside (views/index.ejs)

Kopier inn:

```
<!doctype html>
<html lang="no">
<head>
    <meta charset="utf-8" />
    <title>Nannestad Pizzabar - Ordresøk</title>
</head>
<body>
    <h1>Nannestad Pizzabar - Ordresøk</h1>

    <p><b>Visning:</b> <%= title %></p>

    <% if (error) { %>
        <p style="color:red;"><%= error %></p>
    <% } %>

    <form method="POST">
        <label>Ordrenr (ordre_id):</label>
        <input name="ordre_id" placeholder="f.eks. 3" />

        <div style="margin-top:10px;">
            <button formaction="/ordreinfo">Vis ordreinfo</button>
            <button formaction="/produktinfo">Vis produktinfo</button>
            <button formaction="/kundeinfo">Vis kundeinfo</button>
            <button formaction="/allinfo">Vis all info</button>
        </div>
    </form>

    <p style="margin-top:10px;">
        <a href="/alleordre">Gi alle ordre</a> (GET)
    </p>

    <hr />

    <% if (rows && rows.length > 0) { %>
        <table border="1" cellpadding="6">
            <thead>
                <tr>
                    <% for (const key of Object.keys(rows[0])) { %>
                        <th><%= key %></th>
                    <% } %>
                </tr>
            </thead>
            <tbody>
                <% for (const r of rows) { %>
                    <tr>
                        <% for (const key of Object.keys(rows[0])) { %>
                            <td><%= r[key] %></td>
                        <% } %>
                    </tr>
                <% } %>
            </tbody>
        </table>
    <% } else if (rows) { %>
```

```
<p>Ingen treff.</p>
<% } else { %>
  <p>Skriv inn et ordrenr og trykk en knapp, eller trykk "Gi alle ordre".</p>
<% } %>
</body>
</html>
```

Del 6: Kjør og test

Start server

```
node server.js
```

Gå til: <http://localhost:3000>

Test:

- Skriv inn **1** → trykk **Vis ordreinfo**
- Skriv inn **1** → trykk **Vis kundeinfo**
- Skriv inn **1** → trykk **Vis produktinfo**
- Skriv inn **1** → trykk **Vis all info**
- Trykk **Gi alle ordre** (lenken) → skal vise kort liste for alle ordre