Oracle + n8n Workflow Setup Guide

1. Run Oracle XE in Docker

```
docker run -d \
--name oracle-xe \
--platform linux/amd64 \
-p 1521:1521 \
-e ORACLE_PWD=MyPaSSword@123 \
gvenzl/oracle-xe
```

Check readiness:

```
docker logs oracle-xe | grep -i "DATABASE IS READY"
```

2. Start n8n in Docker

```
docker run -d \
--name n8n \
-p 5678:5678 \
-v ~/.n8n:/home/node/.n8n \
n8nio/n8n
```

Open: http://localhost:5678

3. Create Oracle Credential in n8n

Host: host.docker.internal

Port: 1521

Use Service Name: Yes

Service Name: XEPDB1

User: system

Password: MyPaSSword@123

4. Test Connection

Run query:

```
SELECT 'Hello from Oracle + n8n' AS message FROM dual;
```

5. Create Customers Table

```
CREATE TABLE customers (

customer_id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

first_name VARCHAR2(50),

last_name VARCHAR2(50),

email VARCHAR2(100),

created_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

status VARCHAR2(20),

total_orders NUMBER DEFAULT 0,

last_purchase_date TIMESTAMP

);
```

6. Insert Sample Data (PL/SQL Block)

```
BEGIN

INSERT INTO customers (first_name, last_name, email, created_date, status, total_orders,
last_purchase_date)

VALUES ('John','Doe','john.doe@email.com',SYSTIMESTAMP - INTERVAL '90'
DAY(3),'active',15,SYSTIMESTAMP - INTERVAL '5' DAY(3));

INSERT INTO customers
(first_name,last_name,email,created_date,status,total_orders,last_purchase_date)

VALUES ('Jane','Smith','jane.smith@email.com',SYSTIMESTAMP - INTERVAL '120'
DAY(3),'active',23,SYSTIMESTAMP - INTERVAL '2' DAY(3));

INSERT INTO customers
(first_name,last_name,email,created_date,status,total_orders,last_purchase_date)

VALUES ('Bob','Johnson',NULL,SYSTIMESTAMP - INTERVAL '30' DAY(3),'inactive',0,NULL);
```

COMMIT;

END;

7. Data Quality Query

SELECT 'Missing Email' AS issue_type,

COUNT(*) AS count,

LISTAGG(first_name || ' ' || last_name, ', ') WITHIN GROUP (ORDER BY first_name) AS affected_records

FROM customers WHERE email IS NULL

UNION ALL

SELECT 'Inactive with Orders', COUNT(*),

LISTAGG(first_name || ' ' || last_name, ', ') WITHIN GROUP (ORDER BY first_name)

FROM customers WHERE status='inactive' AND total_orders > 0

UNION ALL

SELECT 'Stale Active Customers', COUNT(*),

LISTAGG(first_name || ' ' || last_name, ', ') WITHIN GROUP (ORDER BY first_name)

FROM customers WHERE status='active'

AND (last_purchase_date IS NULL OR last_purchase_date < SYSTIMESTAMP - INTERVAL '30' DAY(3));

8. Add IF Node Logic

Expression:

{{ $input.all().filter(item => item.json.COUNT > 0).length }}

9. Gmail Alert Node

Message (Expression):

{{

'Data Quality Issues Detected:\n\n' +

$('Check Data Quality').all()

```
.filter(i => i.json.count > 0)
.map(i => {
const r = i.json;
return [
`Issue: ${r.issue_type}`,
`Count: ${r.count}`,
`Affected: ${r.affected_records}`,
'--------------------'
].join('\n');
}).join('\n')
}}
```

This workflow will email you only when issues exist.