

WHERE...

- ☐ Standardabweichung der Gehälter aller Mitarbeiter:

```
SQL> SELECT STDDEV(gehalt) FROM MITARBEITER;
```

Gruppierung von Datensätzen:

Group by, Datensätze lassen sich mit Hilfe der Anweisung **GROUP BY** nach einem zu spezifizierendem Kriterium gruppieren.

```
SELECT <attributname_1, attributname_2...> FROM <tablename>
GROUP BY <column>;
```

- ☐ Beispielsweise kann eine Aufstellung der Anzahl gestellter Rechnungen pro Datumstag vorgenommen werden.

```
SQL > SELECT datum, COUNT(*) FROM RECHNUNG GROUP BY datum;
```

Spezielle Datenbankoperatoren:

Having

- ☐ Ermittlung der durchschnittlichen Gehälter der Berufsgruppen. Aus statistischen Gründen sollen nur Gruppen, die mindestens vier Tupel beinhalten, ausgewertet werden.

```
SQL> SELECT position, AVG(gehalt) FROM MITARBEITER GROUP BY
position HAVING COUNT(*) > 4;
```

In, geschachtelte Select-Befehle:

- ☐ Ermittlung der Lieferantenummer und den Namen aller Lieferanten, die das Teil No. 3 bereits geliefert haben.

```
SQL> SELECT lieferanten.no, name FROM LIEFERANT
WHERE lieferanten.no IN (SELECT lieferanten.no FROM BESTELLUNG
WHERE teil.no=3);
```

- ▷ Die Verschachtelung über die In-Klausel kann beliebig tief erfolgen.

Exists

- ☐ Ermittle die Namen der Lieferanten, die alle Teile liefern.

```
SQL> SELECT lieferantenname FROM LIEFERANT l WHERE NOT EXISTS
(SELECT * FROM TEILE t WHERE NOT EXISTS (SELECT * FROM
BESTELLUNG b WHERE lieferanten.no=l.lieferanten.no AND
teile.no=t.teile.no));
```

Anlegen einer Relation aus bereits in der Datenbank existierenden Daten:

Das SQL-Kommando **CREATE** kann auch zum Anlegen einer Relation aus bereits in der Datenbank existierenden Daten verwendet werden.

```
CREATE TABLE <table> AS SELECT ...
```

- ▷ Der Select-Teil des Create-Kommandos kann dabei alle Basisoperationen wie Projektion, Selektion, Join, Union usw. beinhalten.

d) Views

View (Sicht), virtuelle Tabelle, bei der die Daten nicht physisch gespeichert werden, sondern der SQL-Befehl zur Erzeugung der View gespeichert wird. Die Generierung der Relation erfolgt dynamisch zur Laufzeit.

- Die Generierung von Views erfolgt analog zur Erzeugung von Relationen:

```
CREATE VIEW <view> AS SELECT ...
```