Zusammengefaßte SQL-Referenz:

1 Allgemeine Struktur der Sprache:

DDL (Data Definition Language): Anweisungen zur Datendefinition

DML (Data Manipulation Language): Selektion und Update der Daten.

DCL (Data Control Language): Sicherheit. Vergabe von Zugriffsrechten. Integrität (Korrektheit/Vollständigkeit) der Daten.

Die Grundbefehle der drei Sprachen stehen immer am Anfang einer SQL-Abfrage. Dann erst folgen Attribute, Datentypen, Operatoren und Funktionen.

2 Allgemeiner Aufbau der Datenbanken:

Tabelle (Table): Besteht aus Zeilen und Spalten.

Spalte (Column): Attribute der Entität.

Zeile (Row): Datensätze.

Wert (Value): Schnittmenge aus Spalte und Zeile.

Sicht (View): Bildschirmausgabe der Tabelle (-nkombinationen).

Index: Speicher der Schlüssel- und Positionsangaben => Beschleunigung Zugriff, Erzwingung Eindeutigkeit.

Gruppierung (Cluster): Strukturierung der Daten und Performance-Verbesserung.

2.1 Wichtige Begriffe:

Schlüssel (Key): Mittel zur eindeutigen Bezeichnung von Datensätzen.

- **Primärschlüssel (Primary Key):** Eindeutige Bezeichnung einer Zeile durch eine oder mehrerer Spalten (dann: Composite Primary Key).

- **Fremdschlüssel (Foreign Key):** Eindeutige Bezeichnung von Relationen zwischen verschiedenen Tabellen. Fremdschlüssel einer Tabelle werden von den primärschlüsseln anderer Tabellen gebildet.

- **Eindeutiger Schlüssel (Unique Key):** Wie Primärschlüssel, aber wird in Spalten eingerichtet, die neben dem Primärschlüssel einen weiteren Schlüssel darstellen.

Integritätsbedingungen (Constraints): Überwachung der Integrität der Datenbank, durch die Kontrolle der Relationen zwischen den Tabellen, also der Primär- und Fremdschlüssel. Bei der Erstellung der Tabelle können folgende Integritätsbedingungen festgelegt werden.

Not Null-Restriktion: Erzwingt Einfügung eines Wertes in die Spalte.

Unique-Restriktion: Erzwingt Eindeutigkeit eines Wertes in der Spalte.

Primary Key: Beinhaltet die Eigenschaften Unique und Not Null für eine oder mehrere Spalten.

Foreign Key: Die aufgeführte Spalte muß sich an dem Primärschlüssel einer anderen Tabelle orientieren.

Check: Zeilenwerte müssen Vorgaben genügen (z.B. Gehaltsbeschränkung, Boolesche Operatoren).

2.2 Datentypen:

BFILE: Ermöglicht den Zugriff über das DBS auf Daten im Filesystem, außerhalb des DBS.

BLOB: Speichert Binary Large Objects. Keine Beschränkung auf maximal eine Spalte, wie bei Long-Datentyp.

BYTE: Speichert binäre Dateien (Bilder, Sprachmuster). Mustererkennung mit LIKE ist hier nicht möglich.

CHAR(n): Speichert Zeichenfolge mit der maximalen Länge n. Zulässige Länge hängt vom DBMS ab.

CHARACTER(n): Synonym für CHAR.

CHARACTER VARYING: Speicherung variabel langer Zeichenkette. In Oracle und Informix ist VARCHAR gebräuchlich.

CLOB: Speichert Character Large Objects (übergroße Zeichenketten, max. 4 GB). Nur bei Oracle.

LONG: Speichert Daten vom Typ VARCHAR bis zu 2 GB. Nur bei Oracle.

LONG VARCHAR: Wie LONG.

NCHAR, NVARCHAR: Speichert Zeichenketten einzelner Spalten in einer anderen Codierung z.B. anderer nationaler Zeichensatz).

RAW(n): Speichert binäre Daten. Nur bei Oracle.

LONG RAW: Speichert binäre Daten bis 2 GB Größe. Nur bei Oracle.

ROWID: Identifiziert eine Zeile der Tabelle eindeutig und benennt die Position in der DB in auswertbaren Hexadezimalwerten. Spalten vom Typ ROWID können nicht vom Benutzer definiert werden. Nur bei Oracle.

SERIAL: Enthält eine eindeutige und fortlaufende Zahl, die vom DBMS automatisch generiert wird. Anfangswert ist 1. Benutzer kann Zahl nicht manipulieren, aber neue durch Zeileneinfügung hervorrufen. Nur Informix, bei Oracle über CREATE SEQUENCE).

TEXT: Wie CLOB für Oracle. Nur Informix.

VARCHAR(n): Speichert Zeichenkette variabel. Schwanken die Längen der Zeichensätze, dann ist dieser Typ CHAR vorzuziehen.

VARCHAR2(n): Wie VARCHAR. Nur bei Oracle.

DATE: Speichert Datum (Informix) oder Datum und Uhrzeit (Oracle).

DATETIME: Wie DATE (bei Oracle) für Informix.

INTERVAL: Eingabe von Zeitintervallen. Nur Informix.

DEC(m(n)): Synonym für DECIMAL.

DECIMAL(m(n)): Speichert Zahl mit definierter Größe und Nachkommastellenzahl.

DOUBLE PRECISION: Definiert Gleitkommazahl wie FLOAT.

FLOAT(n): Definiert Gleitkommazahl mit hoher Genauigkeit (Informix 8 Bytes). Entspricht DOUBLE in Programmiersprache C.

INT: Synonym für INTEGER.

INTEGER: Spalte enthält eine ganze Zahl.

MONEY(m(n)): Speichert Währungsbeträge. Nur Informix.

NUMBER(m(n)): Ähnlich DECIMAL, aber ohne Standard. Datentyp für alle numerischen Zwecke. Nur bei Oracle.

NUMERIC(m(n)): Synonym für Festkommazahl unter DECIMAL.

REAL: Repräsentiert Gleitkommazahl und ist Synonym für SMALLFLOAT.

SMALLFLOAT: Repräsentiert Gleitkommazahl mit "einfacher Genauigkeit" 4 Bytes.

SMALLINT: Speichert ganze Zahl.

Konvertierungen:

Nach von	CHAR	NUMBER	DATE
CHAR	----------	TO_NUMBER	TO_DATE
NUMBER	TO_CHAR	----------	TO_DATE
DATE	TO_CHAR	nicht zulässig	----------

Syntax:

CHAR/VARCHAR: '[char]...'

INTEGER: [+|-]ziffer[ziffer]...[K|M] K=KBytes/M=MBytes.

NUMBER: [+|-]ziffer[ziffer]...[Exponent|Multiplizierer]

Exponent: {e|E}[+|-]ziffer[ziffer]...

Multiplizierer: K oder M

DATE:

Jahrhundert	Jahr	Monat	Tag
Stunde	Minute	Sekunde	

Defaultwert Zeit: 12:00:00 a.m.

Defaultwert Datum: Erster Tag des laufenden Monats.

Aktuelle Werte: Liefert SYSDATE (Maschinendatum).

Konstanten vom Typ DATE: Stehen in Hochkommata '24-DEC-98'.

Rechnungen mit DATE: Datum-Datum; Datum +/- Konstante (z.B. SYSDATE-7).

2.3 Operatoren:

(...)	Überschreibt die normalen Vorrangregeln.
+ -	Bezeichnet einen positiven oder negativen Ausdruck.
* / + -	Multiplizieren, dividieren, addieren, subtrahieren.
||	Konkatenation von Character-Werten.
=	Test auf Gleichheit.
!=, ^=,<>	Test auf Ungleichheit.
<, >, <=, >=	Tests auf kleiner/größer als.
IN	Gleichheit zu einem Mitglied einer Menge.
NOT IN	Äquivalent zu "!=ALL"

ANY	Vergleicht einen Wert mit jedem Wert aus einer Liste oder Subquery. Vorausgehen muß einer dieser Operatoren: =, !=, <, >, <=, >=
ALL	Vergleicht einen Wert mit allen Werten aus einer Liste oder Subquery. Vorausgehen mmuß einer dieser Operatoren: =, !=, <, >, <=, >=
[NOT]BETWEEN x AND y	[Nicht] größer/gleich x und kleiner/gleich y. Können auch Zeichenketten sein.
[NOT]EXISTS	Liefert TRUE, wenn eine Subquery mindestens [keine] eine Zeile zurückliefert.
[NOT]LIKE	Stimmt [nicht] mit folgendem Muster überein: Zeichen "%" entspricht irgendeinem String der Länge >=0, "_" entspricht genau einem Zeichen.
IS[NOT]NULL	Test auf NULL-Wert.
NOT	Invertiert Ergebnis eines logischen Ausdrucks.
AND	Logisches UND.
OR	Logisches ODER.
UNION	Kombiniert Queries, indem alle Zeilen geliefert werden, die von jeder einzelnen Abfrage erfaßt werden (Vereinigung).
INTERSECT	Mengendurchschnitt der selektierten Zeilen.
MINUS	Mengendifferenz zwischen zwei selektierten Mengen.
COUNT(expr)	Liefert die Anzahl der Zeilen, für die der Ausdruck expr nicht NULL ist.
COUNT(*)	Liefert alle Zeilen einer Tabelle.
DISTINCT	Eliminiert doppelte Zeilen oder doppelte Werte in einem Aggregatausdruck.

2.4 Funktionen:

EXP(n)	Exponentialfunktion
LN(n)	Natürlicher Logarythmus
LOG(m,n)	Logarythmus von n zur Basis m
POWER(m,2) n=2	Allgemeine Potenzfunktion m^2
COS(n)	Trigonometrische Funktion Cosinus. Argumente im Bogenmaß.
SIN(n)	Trigonometrische Funktion Sinus. Argumente im Bogenmaß.
TAN(n)	Trigonometrische Funktion Tangens. Argumente im Bogenmaß.
ACOS(n)	Trigonometrische Umkehrfunktion Cosinus. Argumente im Bogenmaß.
ASIN(n)	Trigonometrische Umkehrfunktion Sinus. Argumente im Bogenmaß.

ATAN(n)	Trigonometrische Umkehrfunktion Tangens. Argumente im Bogenmaß.
COSH(n)	Hyperbolische Funktion Hyperbelcosinus.
SINH(n)	Hyperbolische Funktion Hyperbelsinus.
TANH(n)	Hyperbolische Funktion Hyperbeltangens.
CEIL(n)	Konvertierung in die nächsthöhere ganze Zahl.
FLOOR(n)	Konvertierung in die nächstniedrigere ganze Zahl.
ROUND(n,[m])	Runden. Wenn m negativ ist, wird n auch links vom Dezimalpunkt gerundet.
TRUNC(n,[m])	Abschneiden. Wenn m negativ ist, wird n auch links vom Dezimalpunkt abgeschnitten.
ABS(n)	Absolutwert.
MOD(m,n)	Modulo-Funktion.
SIGN(n)	Signum-Funktion
SQRT(n)	Wurzelfunktion.
CHR(n)	Konvertiert Argument n in das dementsprechende Zeichen. Grundlage ist der Zeichensatz, der bei der DB-Erstellung angegeben wurde.
CONCAT(s1,s2)	Konkateniert zwei Zeichenketten, wie Operator "||"
INITCAP(s)	Konvertiert eine Zeichenkette, indem der Anfangsbuchstabe eines Wortes groß und die übrigen Zeichen klein ausgegeben werden.
LOWER(s)	Konvertiert eine Zeichenkette vollständig in Kleinbuchstaben.
LPAD(s,n[,fs])	Der String s wird auf die Länge n gebracht, indem s von links mit den Zeichen aus dem Füllstring fs gefüllt wird. Standardmäßig ist fs ein einzelnes Blank. Sinnvoll bei Darstellung von Hierachien.
RPAD(s,n[,fs])	Bringt den String s auf die Länge n, indem s von rechts mit den Zeichen aus dem Füllstring fs gefüllt wird. Standardmäßig ist fs ein einzelnes Blank. Sinnvoll bei Darstellung von Hierachien.
REPLACE(s,ss[,rs])	Ersetzt den Suchstring (ss) durch den Ersetzungsstring (rs), falls er im String s gefunden wird. Wird rs weggelassen, wird ss aus s entfernt.
SUBSTR(s,m[,n])	Liefert den Teilstring des Strings s ab der Stelle m (ab 1) in der Länge n zurück. Wird n weggelassen, erhält man den Teilstring ab Position m bis zum rechten Ende von s.
TRANSLATE(s,t1,t2)	Ermöglicht die komplexe Codierung von Zeichen. Jedes Zeichen in der Zeichenkette t1 wird in s durch das Zeichen aus t2 ersetzt, das an derselben Position steht.
UPPER(s)	Konvertiert eine Zeichenkette vollständig in Großbuchstaben.
ASCII(c)	Konvertiert ein Zeichen in den entsprechenden Code des zugrundeliegenden

	Zeichensatzes (z.B. ASCII).
INSTR(s,ts[,n[,m]])	Liefert Position des m-ten Auftretens des Teilstrings ts im String s ab der Position n (oder ab 1).
LENGTH(s)	Ermittelt die tatsächliche Länge der Zeichenkette s.
AVG([DISTINCT|ALL]n)	Liefert den Mittelwert über die numerische Spalte n, ohne NULL-Werte und DISTINCT eliminiert Duplikate.
COUNT([DISTINCT|ALL]expr) **COUNT (*)**	Zählfunktion. Bereits bei Operatoren aufgeführt.
MAX([DISTINCT|ALL]n) **MIN([DISTINCT|ALL]n)** **STDDEV([DISTINCT|ALL]n)** **SUM([DISTINCT|ALL]n)** **VARIANCE([DISTINCT|ALL]n)**	Liefert den höchsten (MAX), niedrigsten (MIN) Wert, die Standardabweichung (STDDEV), die Summe (SUM) und die Varianz der Werte (VARIANCE) über die numerische Spalte n, ohne NULL-Werte und DISTINCT eliminiert Duplikate.
CHARTOROWID(s) **ROWIDTOCHAR(r)** **HEXTORAW(s)** **RAWTOHEX(r)**	Führen eine sogenannte Cast-Operation aus, damit die konvertierten Werte z.B. in Vergleichs- oder Zuweisungsoperationen kompatibel zu anderen involvierten Typen sind. Hexadezimale Werte werden in Hochkommata eingeschlossen und damit wie Zeichenketten behandelt.
TO_CHAR(n[,f]) **TO_CHAR(n[,f])**	Ermöglicht die optional mit dem Formatstring f gesteuerte Umwandlung der Zahl n oder des Datums d in eine Zeichenkette.
TO_DATE(s[,f])	Wandelt eine Zeichenkette s, optional mit dem Formatstring f gesteuert, in ein Datum um; die Anzeige des Datums hängt vom Standardformat für Datumstypen ab.
TO_NUMBER(s[,f])	Wandelt eine Zeichenkette s, optional mit dem Formatstring f gesteuert, in eine Zahl um.
LAST_DAY(d)	Gibt den letzten Tag des Monats an, in dem das Datum d liegt.
MONTH_BETWEEN(d1,d2)	Differenz des Datums d1 zum Datum d2 in Monaten.
GREATEST(expr[,expr]...)	Liefert aus der angegebenen Liste der Ausdrücke expr denjenigen mit der lexikographisch höchsten Ordnung. Hängt vom Zeichensatz der DB ab.
LEAST(expr[,expr]...)	Liefert aus der angegebenen Liste der Ausdrücke expr denjenigen mit der lexikographisch niedrigsten Ordnung. Hängt vom Zeichensatz der DB ab.

NVL(expr1,expr2)	Ersetzt den Ausdruck expr1 durch den Ausdruck expr2, wenn expr1 einen NULL-Wert hat.

2.5 Befehle der Datendefinition (DDL):

Befehle und Attribute zur Einrichtung, Modifikation oder Entfernung von Objekten aus der Datenbank. Es werden "leere" Strukturen geschaffen, also die Grundgerüste der DB. In der Datenmanipulation (DML) werden dann die konkreten Inhalte geschaffen, verwaltet und selektiert. Die Befehle der Datendefinition sind in der Regel dem DBA vorbehalten.

2.5.1 Grundbefehle:

CREATE: Erstellt Objekte innerhalb der DB.

ALTER: Ändert bereits erstellte Objekte innerhalb der DB..

DROP: Entnimmt bereits erstellte Objekte aus der DB.

2.5.2 Attribute:

CLUSTER: Erstellung, Änderung, Entfernung eines Datenverbundes mehrerer Tabellen, die eine hohe Beziehungsintensität aufweisen (z.B. bei anhaltenden ähnlichen Abfragen).

DATABASE: Erzeugung, Änderung, Entfernung einer Datenbank(struktur).

INDEX: Erzeugung, Änderung, Entfernung eines Index für eine Tabelle oder einen Cluster. Dient der Verbesserung der Zugriffsmöglichkeiten auf häufig aufgerufene Spalten.

PROFILE: Erzeugung, Änderung, Entfernung eines Profils, mit diversen Einstellungen und Limitierungen, die unter einem bestimmten Namen definiert werden.

ROLLBACK SEGMENT: Erzeugung, Änderung, Entfernung eines DB-Abschnitts, der Informationen über die Arbeitsschritte der Datenbanknutzung speichert, und damit ein Zurückrollen der letzten Schritte erlaubt. Die Parameter hierfür können detailliert festgelegt werden.

SEQUENCE: Anweisung zur Erzeugung, Änderung, Entfernung sogenannter Sequenzen zur Generierung von durchgehenden Werteketten (z.B.Angestelltennummern). Die einfachste Form ist ein normaler Zähler.

SESSION: (Nur mit dem ALTER-Befehl). Änderung zahlreicher Parameter während einer aktuellen Sitzung.

SNAPSHOT: Erzeugung, Änderung, Entfernung der Charakteristika eines Snapshots (Refresh-Zeit, Refresh-Modus, Snapshot-Speicherparameter).

SYNONYM: (Nur mit dem CREATE- und DROP-Befehl). Erzeugung oder Entfernung eines Synonyms. Synonyme machen Sinn, wenn ein Datenbank-Objekt unter mehreren Namen zugänglich sein soll (z.B. unter technischer oder umgangssprachlicher Bezeichnung).

SYSTEM: (Nur mit dem ALTER-Befehl). Ermöglicht das Ändern der Oracle-Instanz durch Aufruf spezieller Funktionen.

TABLE: Erzeugung, Änderung, Entfernung einer Tabellenstruktur. Einflußnahme auf die Wertebereiche, Integritätsbestimmungen und Namen der Spalten. Festlegung etc. der Speicherlokation und anderer Parameter zur Speicherung.

TABLESPACE: Erzeugen, ändern und löschen eines Platzes in der DB zur Speicherung von Schema-Objekten, Rollback-Segmenten und temporären Segmenten. Angabe, Änderung, Entfernung der Datenfiles (mit DATAFILE), aus denen der Tablespace besteht. Mit dem Zusatz ONLINE macht man den Tablespace sofort für alle berechtigten Anwender zugänglich.

TRIGGER: (Nur mit dem CREATE-Befehl). Erzeugung eines gespeicherten PL/SQL-Blocks, der mit einer Tabelle relational zusammenhängt. Damit kann der Nutzer Verarbeitungs- und Prüfungsroutinen aus den Anwendungen heraus in die Datenbank verlagern. Dieser sogenannte Trigger wird automatisch gestartet, wenn eine SQL-Abfrage auf die jeweilige Tabelle abzielt.

TYPE: (Nur mit dem CREATE-Befehl). Ermöglicht die Definition eigener Objekttypen und die Angabe der Prozeduren (Methoden), die zu deren Bearbeitung nötig sind. Grundlage der objektorientierten Eigenschaften in Oracle 8. Zudem sind so verschachtelte Tabellen zu kreieren.

USER: Erzeugung, Änderung, Entfernung eines Datenbank-Nutzers in der DB (Anfangspasswort, wo seine Objekte gespeichert sind, wieviel Ressourcen er verbrauchen darf).

VIEW: Erzeugung, Änderung, Entfernung von Benutzeransichten auf selektierte Daten. Also eine Ableitung externer Sichten aus dem konzeptionellen Schema.

2.6 Befehle der Datenmanipulation (DML):

Hier werden die Inhalte in die DB-Strukturen eingefügt, bearbeitet und ggf. wieder gelöscht. Zudem werden in der Datenmanipulation mit dem SELECT-Befehl auch die Abfragen erzeugt, die ja einer Datenbank erst den zentralen Sinn ihrer Erstellung und Pflege geben.

2.6.1 Grundbefehle:

DELETE: Befehl zum Löschen von Datensätzen in die Datenbankstrukturen.

TRUNCATE: Befehl zum Löschen von Zeilen in Tabellen und Clustern und Freigabe des dort verwendeten Speicherplatzes. Nur bei Oracle.

INSERT: Befehl zum Einfügen von Datensätzen in die Datenbankstrukturen.

UPDATE: Befehl zur Aktualisierung von Datensätzen.

SELECT: Abfragebefehl, mit dem der Benutzer eine bestimmte Sicht auf die angeforderten Datensätze erhalten kann. (Bei der Abfrage SELECT ist unbedingt die Klausel FROM erforderlich).

2.6.2 Klauseln:

FROM: Klausel zur Angabe der Tabelle, aus der die Datensätze selektiert werden sollen (Angabe der Lokation).

WHERE: Klausel zur Einschränkung der Selektion auf ganz konkrete Daten durch Operatoren etc. (Angabe von Bedingungen).

2.6.3 Attribute:

***:** "Wildcard", die dafür sorgt, daß alle Datensätze der Spalten angezeigt werden.

START WITH / CONNECT BY: Sortiert die Reihen hierarchisch.

GROUP BY: Gruppiert die selektierten Zeilen nach dem Wert der Ausdrücke und liefert eine einzelne Zeile an Informationen je Gruppe.

HAVING: Funktionsumfang wie die WHERE-Klausel, wird in Verbindung mit GROUP BY eingesetzt.

ORDER BY: Sortiert das Selektionsergebnis nach den angegebenen Spalten (ASC aufsteigend, DESC absteigend).

FOR UPDATE: Sperrt die selektierten Reihen für eine Aktualisierung.

NOWAIT: Benutzer erhält Kontrolle zurück, wenn eine Sperre nicht erworben werden konnte.

2.6.4 Subqueries:

In der Retrievalsprache SQL gibt es die Möglichkeit, SELECT-Befehle zu verschachteln. Diese Verschachtelungen, bei denen sich eine Abfrage auf eine andere Abfrage bezieht, heißen Subqueries. Subqueries werden mittels Klammern eingebunden.

Bsp.: SELECT kundenname, letzter_auftrag FROM kunde

WHERE kundenname IN (SELECT name FROM lieferant)

In einem sogenannten Verbund wäre diese Abfrage ebenfalls möglich gewesen.

Bsp.: SELECT kundenname, letzter_auftrag FROM lieferant, kunde

WHERE lieferant.name = kunde.kundenname

2.7 Befehle der Benutzerklassen und Privilegien (DCL):

Befehlsklasse für die systematische Erstellung, Bearbeitung und Entfernung von sogenannten Rollen. Rollen sind beschriebene und benannte Sammlungen von individuellen und sehr detailliert steuerbaren Rechten. Hier werden Benutzergruppen definiert und maßgeschneiderte Zugriffsrechte vergeben. Die Daten stehen in der Data-Dictionary.

2.7.1 Grundbefehle:

CREATE ROLE: Erzeugt eine neue Rollenbeschreibung in der Data-Dictionary.

ALTER ROLE: Ändert die Autorisierung für den Rollenzugriff.

DROP ROLE: Entfernt eine Rollendefinition.

SET ROLE: Aktiviert/Deaktiviert die Rollenzugehörigkeit zu einer oder mehrerer Rollen während der aktuellen Sitzung. Voraussetzung ist die Zugriffsberechtigung.

GRANT: Einräumung von Zugriffsberechtigungen und Systemprivilegien auf eine Rolle.

REVOKE: Entzieht Zugriffsberechtigungen und Systemprivilegien einer Rolle.

COMMIT: Speichert Änderungen seit dem Beginn von Transaktionen mit SQL-Anweisungen permanent ab.

ROLLBACK: Rollt alle Aktionen einer Transaktion noch einmal zurück zum Beginn dieser oder bis zu einem Savepoint. Beabsichtigtes Rückgängigmachen von Änderungen. Im Fehlerfall (z.B. DB-Absturz) führt das Programm dagegen ein automatisches Rollback durch.

SAVEPOINT: Setzt eine Marke (Savepoint), bis zu der ein Rollback durchgeführt werden kann.

SET TRANSACTION: Setzt Anweisungen für die aktuelle Transaktion.

LOCK TABLE: Sperrt eine Tabelle komplett für alle Benutzer (z.B. während einer generellen Überarbeitung der Gesamttabelle).

2.8 Sonstige SQL-Befehle:

COMMENT: Fügt einen Kommentar in die Data-Dictionary ein, der Bezug zu einer Tabelle, Spalte, einer Ansicht oder einen Snapshot haben kann.

RENAME: Ändert den Namen eines Schema-Objekts. Der Befehl kann für Tabellen, Views, Sequenzen oder private Synonyme verwendet werden.

2.9 SQL-Erweiterungen:

DATABASE: Öffnet eine zugängliche DB und macht sie zur aktuellen DB, teilweise exklusiv (für den DBA).

CLOSE DATABASE: Schließt die mit DATABASE geöffnete DB.

AUDIT: Protokollierung der DB-Aktivitäten bei spezifizierten SQL-Anweisungen oder Operationen auf Schema-Objekten.

NOAUDIT: Deaktivierung des Auditing aus einer vorangegangenen AUDIT-Anweisung.

ANALYZE: Dient bei Objekten wie Indizes, Clustern und Tabellen zum Sammeln von Performance-Statistiken, Validieren von Strukturen und Identifizieren von verketteten Zeilen.

EXPLAIN PLAN: Bestimmt den Ausführungsplan für eine SQL-Anweisung.