Universidad de Granada

decsai.ugr.es

Fundamentos de Bases de Datos

Grado en Ingeniería Informática

Introduccion al SQL: Algunas sentencias adicionales del DML y DDL

DECSAI

Departamento de Ciencias de la Computación e Inteligencia Artificial

Algunas sentencias adicionales
Insert

Sintaxis:

Insert into *tabla* [(*columna*, ...)]{values (*valor*,...) }| *consulta*]

- Permite la inserción dando valores sólo a algunas columnas. Las columnas no mencionadas se rellenan por defecto
- Permite la inserción tupla a tupla (opción "values)
- Permite la inserción "global" (opción "*consulta*") insertando de golpe en la tabla el conjunto de tuplas resultado de la consulta
- En cualquier caso las columnas donde se insertan los datos han de ser compatibles con lo que se inserta.

Ejemplo:

Insert into alumnos_buenos ape1,ape1,nombre,nota
select ape1,ape2,nombre,media from alumnos where media >=7.5;

Algunas sentencias adicionales
Delete

Sintaxis:

Delete tabla [where *condicion*]

- Si se omite la condicion borra todas la tuplas de la tabla
- Si se pone una condición de llave candidata borra una tupla concreta
- La condición puede incluir comparadores de conjunto y ser tan compleja como se quiera

Ejemplo:

Delete asigna where not exists(select * from matricula where asi#=codas);

Elimina aquellas asignaturas que no tienen alumnos matriculados.

Algunas sentencias adicionales

Update

Sintaxis:

Update *tabla* set *columna=expr.* [*columna=exp.* ...] [where *condicion*]

o alternativamente

Update *tabla* set (*columna*[,*columna,* ...]) =(*consulta*) [(*columna*[,*columna*, ...])=(*consulta*)]... [where *condicion*]

- Actualiza las tuplas que verifican la condicion expresada con la misma filosofía que el borrado
- Permite sustituir valores bien con expresiones bien con valores resultantes de consultas, estas pueden ser de cualquier tipo.

Ejemplo:

Update asigna asig set (asig.credt,asig.credpr)=(select max(credt),max(credpr) from asigna where caracter='op')

where asig.carácter='op' and asig.curso='5'

- Actualiza los creditos teoricos y practicos de todas las asignaturas optativas de 5 curso al valor maximo de dichos campos para todas las asignaturas optativas

Algunas sentencias adicionales
Create Index

Sintaxis:

create [unique] index *indice* on {*tabla* (*columna*[asc|desc],[*columna*[asc|desc]] ...) | *cluster*)

[initrans n] [maxtrans n] [tablespace tablespace] [storage storage] [pctfree n] [nosort]

- unique significa que el valor de la clave verifica una condición de unicidad
- nosort significa que no hay que ordenar las filas cuando se crea el indice
- Se pueden crear varios índices por tabla
- Permiten mejorar las consultas cuando se accede a la tabla ordenada según el campo clave del índice y cuando consulta según dicho campo
- Permite crear indices compuestos de hasta 16 componentes
- Por defecto el orden es ascendente
- Los índice pueden ralentizar la actualización de las tablas

Algunas sentencias adicionales
Create Index. Otros tipos de indices

- Indices de clave invertida:

  Invierten el orden de los bytes de la clave. Optimizan el rendimiento del acceso secuencial en configuraciones paralelas de Oracle.

  Sintaxis

  create [unique] index *indice* on {*tabla* (*columna*[asc|desc],[*columna*[asc|desc]] ...) | *cluster*} **reverse**

- Indices de mapa de bits (Bitmap).

  Solo funcionan bien en atributos categóricos y son especialmente útiles cuando el dominio es pequeño

  Sintaxis

  create **bitmap** index *indice* on {*tabla* (*columna*[asc|desc],[*columna*[asc|desc]] ...) | *cluster*}

Algunas sentencias adicionales
Create Index. Otros tipos de indices

- Tablas organizadas por índices

  Son tablas que están organizadas como arboles B de forma que las hojas de los arboles son la tuplas.

  Esta forma de la tabla se debe indicar como una clausula adicional en la sentencia CREATE TABLE

  Sintaxis

  CREATE TABLE *[usuario.]nombre_tabla*

  ({datos_columna | restricciones de tabla}

  [,{datos_columna | restricciones de tabla}]...) **ORGANIZATION INDEX**

  La tabla debe tener especificada una llave primaria

En el cuaderno de prácticas pueden encontrarse ejemplos de uso de todos estos tipo de índices

Algunas sentencias adicionales
Create View

Sintaxis:

create view *vista* [(*alias* [,*alias*] ...)] as *consulta* [with check option [constraint resticcion]]

- Los alias nos permiten renombrar todas las columnas de la vista
- La consulta nos permite construir una visión de usuario tan compleja como queramos. Solo se impide la clausula "order by"
- "with check option" proporciona restricciones adicionales para la actualización mediante vistas
- Una vista puede aparecer en cualquier sentencia "select".
- Una vista puede ser objetivo en una sentencia de actualización; pero hay que tener en cuenta los problemas que la actualización mediante vistas de usuario puede generar.

En el cuaderno de prácticas pueden encontrarse ejemplos de uso de vistas y las restricciones concretas que se impone para su actualización

Algunas sentencias adicionales
Create Cluster

Concepto de Cluster:

- Un "cluster" es una forma de almacenamiento en la que se almacenan juntas la tuplas de distintas tablas que comparten uno o varios campos comunes y se consultan juntas.
- Los cluster se pueden indexar o crear mediante tablas hash

Ejemplo

- Si se van a consultar siempre conjuntamente (sacar listas de alumnos), la tabla asignaturas y la tabla matrícula se pueden almacenar juntas a través del campo código de asignatura.

  Cada ocurrencia de asignatura se almacenaría conjuntamente con las ocurrencias de la tabla matrícula que le corresponden

Algunas sentencias adicionales
Create Cluster indexado

Sintaxis :

CREATE CLUSTER [usuario].cluster

(columna tipo_de_dato [, columna tipo_de_dato]…)

[PCTFREE n], [PCTUSED n], [INITRANS n], [MAXTRAN n] [TABLESPACE nombre], [STORAGE nombre],[SIZE n]

Ejemplo:

```
create cluster listas(asi# varchar(4))
 create table asigna (asi# varchar(4) primary key ,
     nombreas varchar(30) not null,
     curriculum varchar(20) not null ,
     credt number(4,1) not null ,
credpr number(4,1) not null,
caracter char(2) check (caracter in ('tr','ob','op','lc')),
temp char(2) check (temp in ('cu','an')),
    check ((temp='cu' and credt+credpr between 4.5 and 9)
    or (temp='an' and credt+credpr between 6 and 12)))
cluster listas(asi#)
```

Algunas sentencias adicionales
Create Cluster indexado

Ejemplo

```
create table matricula(asi# varchar(4) references asigna,
        codal varchar(8) references alumnos,
        curso varchar(9) not null,
        calificacion char(2) (check calificacion in
           ('np','su','ap','no','sb','mh')),
        primary key (codas,codal,curso))
      cluster listas(asi#)
```

Creación del índice asociado

```
create index idx_listas on cluster listas
```

Algunas sentencias adicionales
Create Cluster hash

Sintaxis :

CREATE CLUSTER *[usuario].cluster*

(columna tipo_dato) [HASH is columna] SIZE <tamaño>

HASHKEY <cantidad_valores_distintos_de _la _clave>

- La clausula HASH se usa cuando la clave de cluster es un valor entero uniformemente distribuido. En caso contrario ORACLE aplica su algoritmo de direccionamiento
- SIZE mide el tamaño en bytes del espacio que van a ocupar las tuplas con del mismo valor de clave:
  - Hay que tener en cuenta las tuplas de las dos tablas
  - Hay que prever si van a haber colisiones
  - Se debe estimar un 15% adidional.
- HASH KEY estima cuantos valores distintos va a tomar la clave del cluster

En el cuaderno de prácticas pueden encontrarse ejemplos de uso de este tipo de cluster y cómo se puede crea una tabla con estructura hash mediante su uso