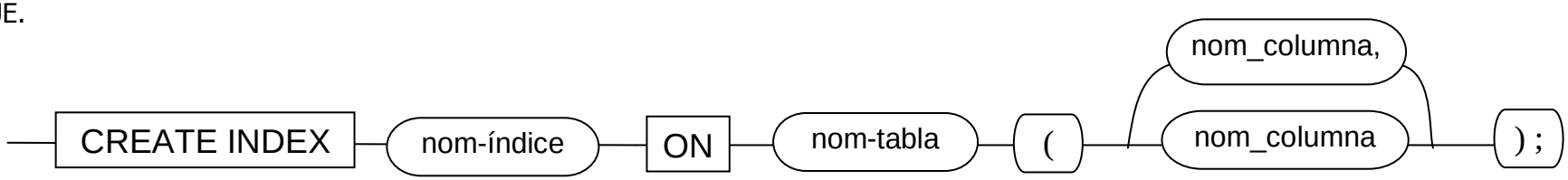
Sesión 13

CREATE INDEX para definir un índice en una tabla

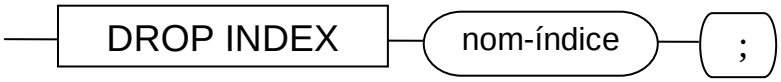
Un índice es una estructura de memoria secundaria que permite el acceso directo a las filas de una tabla. Hace que el tiempo de respuesta a una consulta disminuya, mejorando su rendimiento y optimizando su resultado. Su manejo se hace de forma inteligente. Es el propio Oracle quien decide qué índice se necesita. Por defecto Oracle siempre crea un índice para cada primaria que se defina y para cada restricción UNIQUE.



Ejemplo:

CREATE INDEX ind_prov_cliente ON cliente(provincia);

Para borrar un índice



Cuando se ejecuta una sentencia en Oracle, el módulo "optimizador" analiza varias estrategias y elige una de ellas, no necesariamente la mejor, sino una lo suficientemente buena según un cierto umbral. Por lo tanto, no siempre que tengamos definido un índice se va a utilizar, aunque a lo mejor utilizándolo la estrategia obtenida sería la mejor posible a la hora de ejecutar una determinada acción. Por otro lado, siempre que se haya definido un índice significa que las operaciones de inserción, modificación o borrado en una tabla llevarán un coste adicional de mantenimiento del índice. Por todo lo explicado no se definen índices para todas las columnas de una tabla.

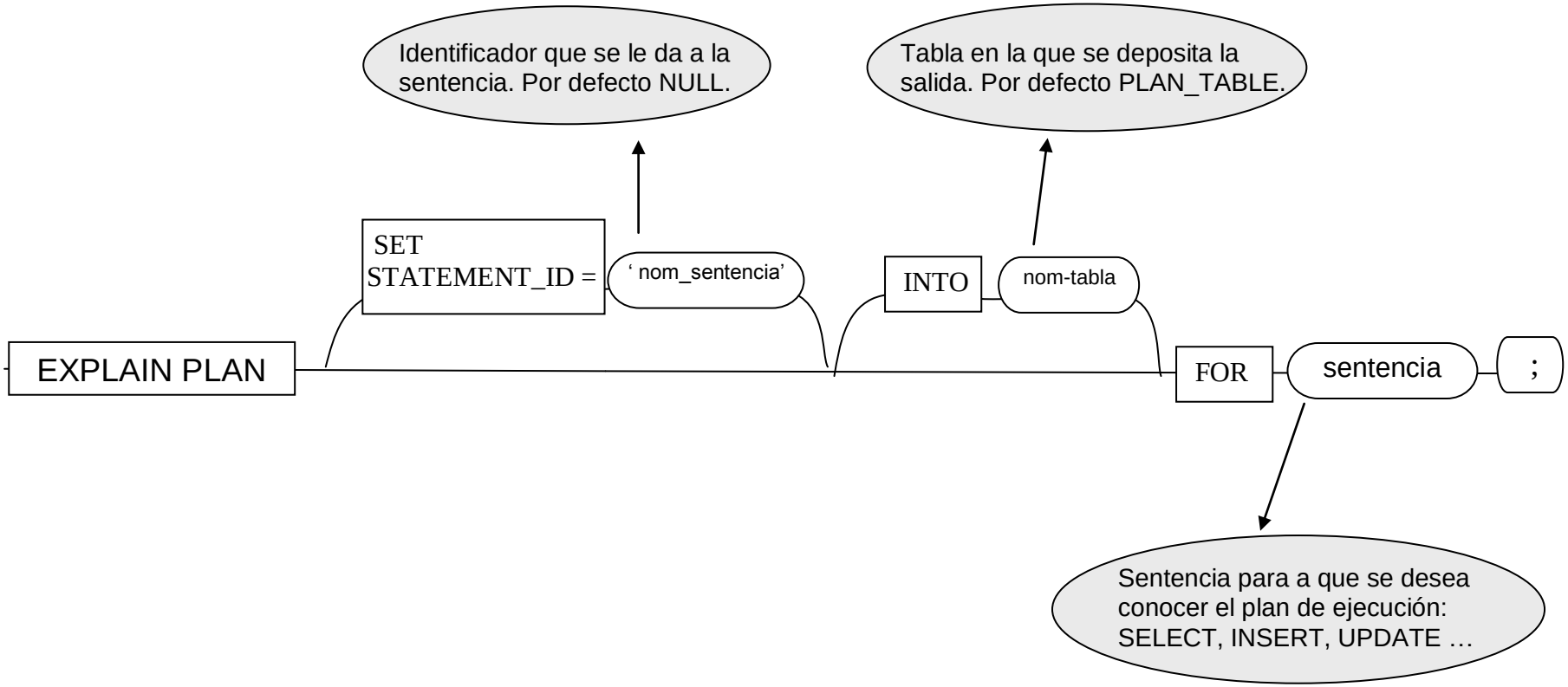
¿Cómo podemos saber si en una consulta se está utilizando, o no, un índice que hemos definido? … Apoyándonos en ***EXPLAIN PLAN***

EXPLAIN PLAN para almacenar el plan de ejecución de una sentencia

Inserta una fila describiendo cada paso del plan de ejecución de una sentencia en la tabla PLAN_TABLE. La sentencia no se llega a ejecutar.

La tabla PLAN_TABLE contiene

- **Statement_id varchar2(30) Valor opcional para identificar planes de ejecución**
- **Operation varchar2(30) Nombre de la operación interna realizada**
- **Options varchar2(225) Detalles sobre la operación**
- **Object_name varchar2(30) Nombre de la tabla o índice**
- **Position numeric**
  - **Para la primera fila indica el coste estimado por el optimizador para realizar la sentencia.**
  - **Para el resto, indica la posición relativa respecto al padre**



Ejemplo:

CASO 1: En la tabla CLIENTE sólo hay un índice definido, el que corresponde a la clave primaria de la tabla. Ejecutamos las siguientes sentencias:

```
 explain plan
 set statement_id='cli1'
for select * from cliente where provincia='Alicante';

select operation, options, object_name, position
from plan_table
where statement_id='cli1'
```

```
OPERATION       OPTIONS         OBJECT_NAME          POSITION
--------------- ------------    ---------------------- ---------------
SELECT STATEMENT                                          3
TABLE ACCESS    FULL            CLIENTE                   1
2 rows selected
```

No se utilizan índices

CASO 2: Si definimos un índice sobre la columna provincia

```
create index ind_prov_cliente  on cliente(provincia);
```

Y ahora ejecutamos

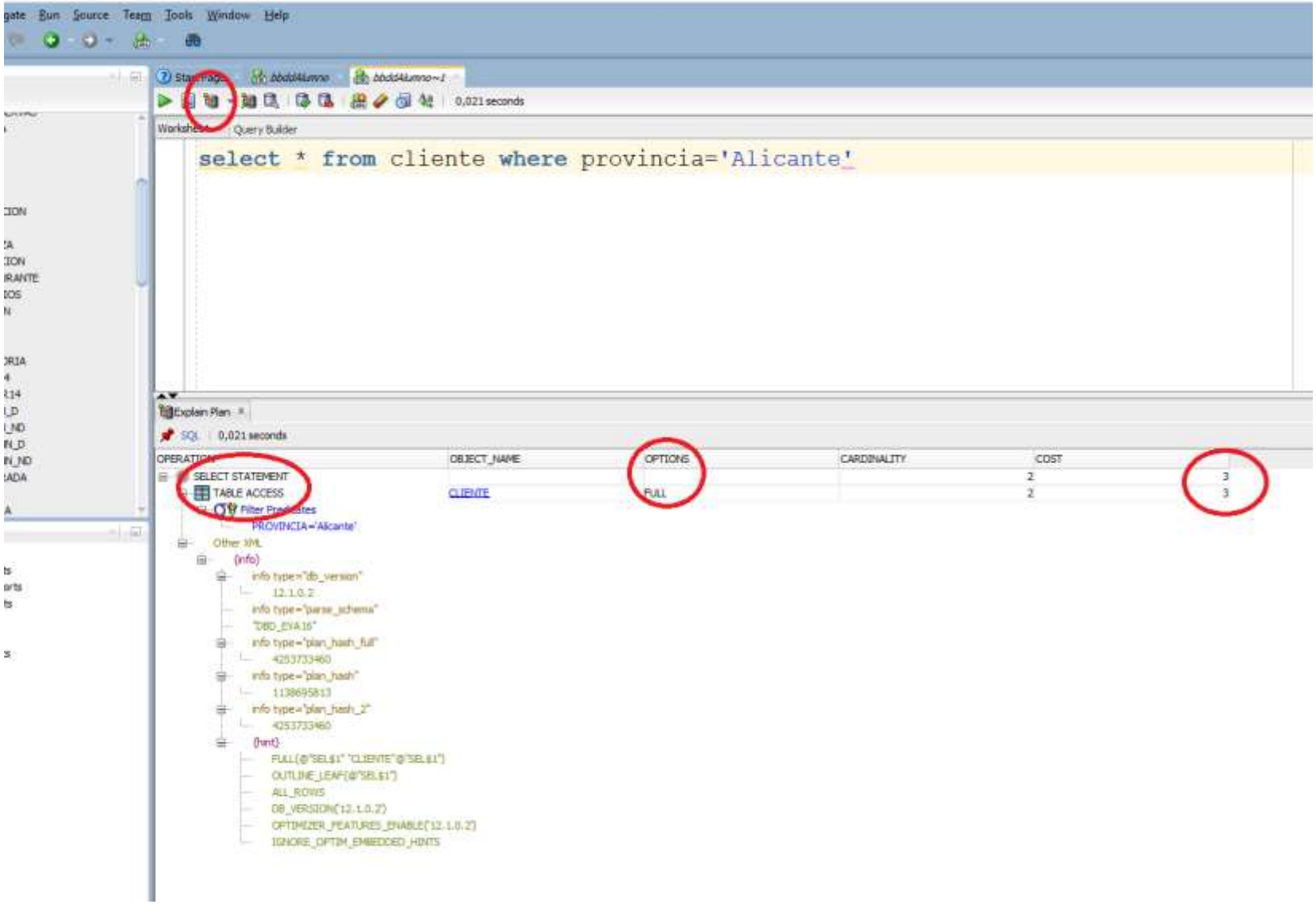
```
explain plan
set statement_id='cli2'
for select * from cliente where provincia='Alicante';

select operation, options, object_name, position
from plan_table
where statement_id='cli2'
```

```
OPERATION       OPTIONS          OBJECT_NAME     POSITION
--------------- -------------    ------------------ --------------
SELECT STATEMENT                                      2
TABLE ACCESS    BY INDEX ROWID   CLIENTE            1
INDEX           RANGE SCAN       IND_PROV_CLIENTE   1
3 rows selected
```

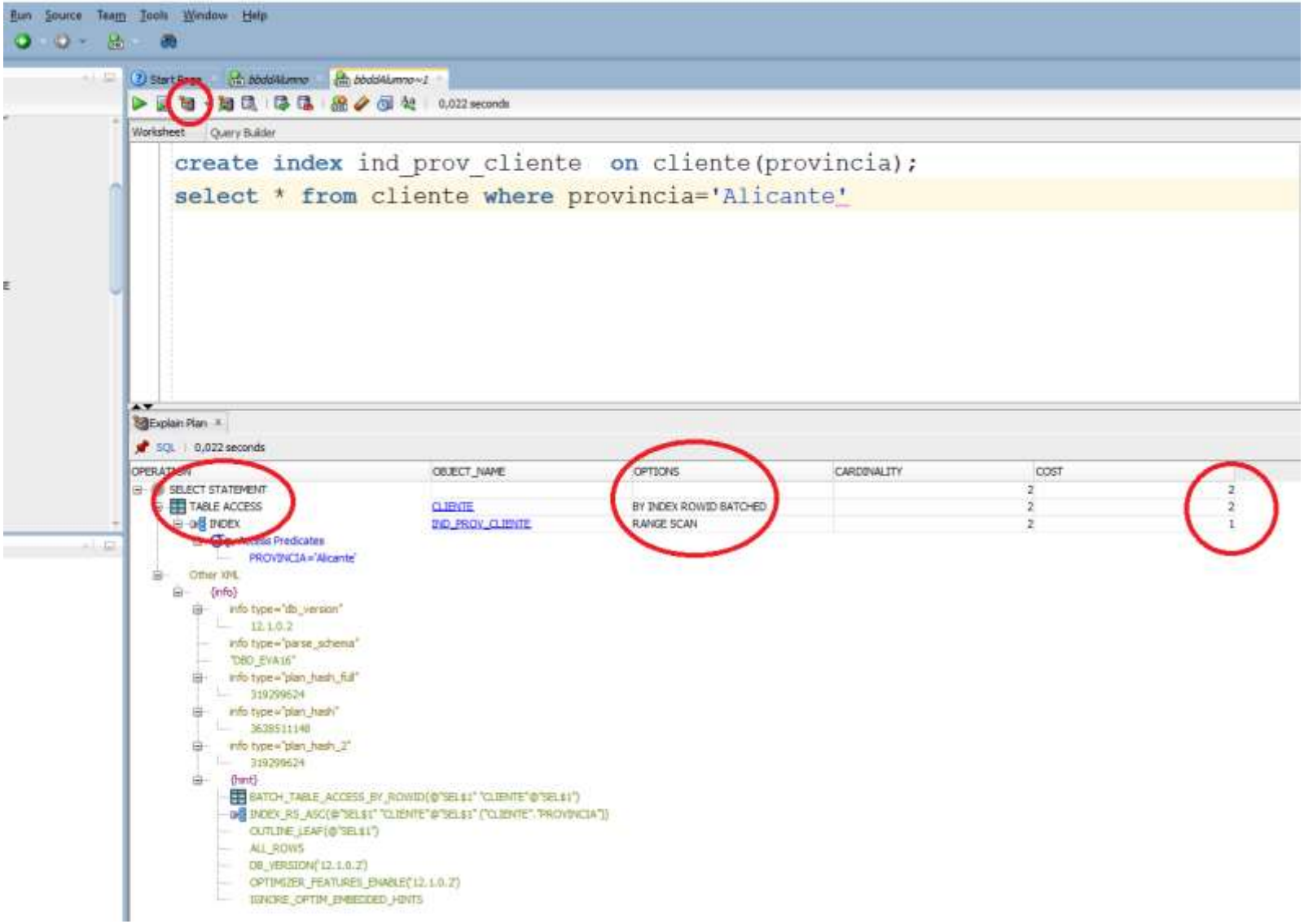
Se utiliza el índice IND_PROV_CLIENTE

Las últimas versiones de SQLDeveloper nos permiten ver el plan de ejecución de manera gráfica

CASO 1: En la tabla CLIENTE sólo hay un índice definido, el que corresponde a la clave primaria de la tabla.

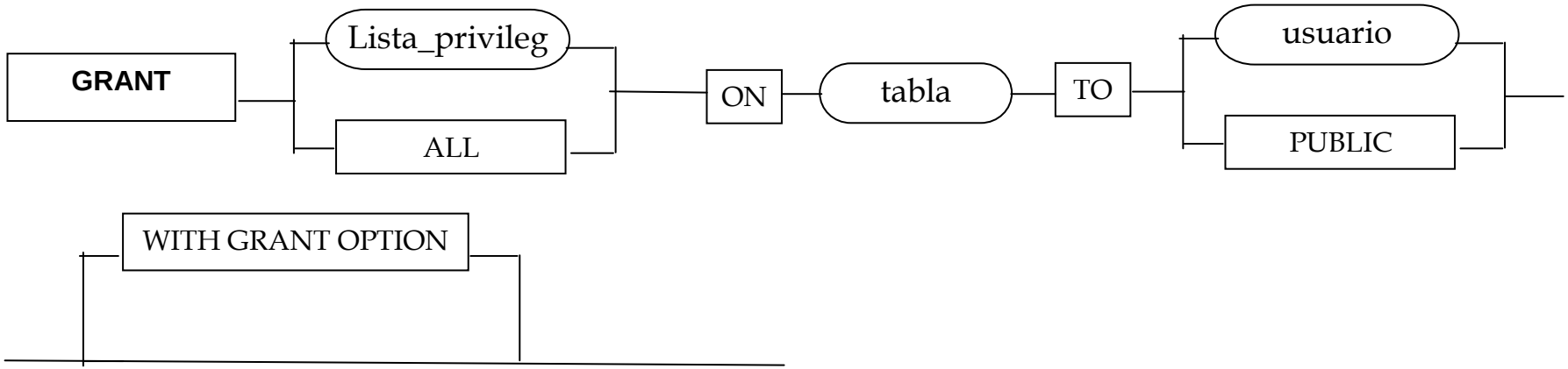


CASO 2: Si definimos un índice sobre la columna provincia.



GRANT

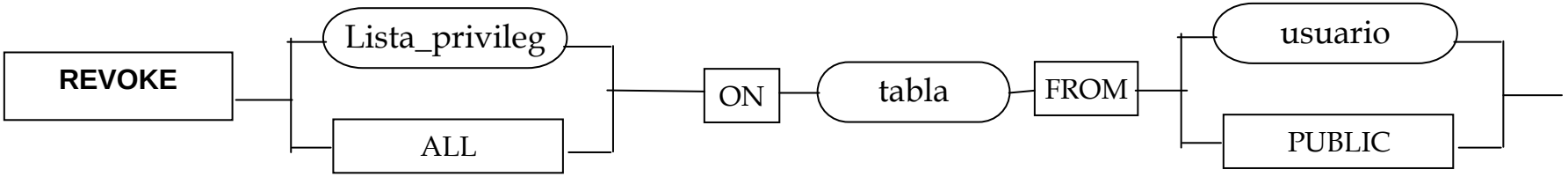
Sirve para otorgar privilegios. Se pueden definir privilegios sobre columnas concretas, y otra serie de opciones que aquí no están contempladas.



Lista de privilegios se refiere a SELECT, INSERT, UPDATE, ALTER, ...
Si ponemos la opción WITH GRANT OPTION, el usuario que recibe los privilegios los puede otorgar a su vez a otros usuarios.
Si en lugar de pober un nombre de usuario ponemos PUBLIC, se le otorgan a todos los usuarios.

REVOKE

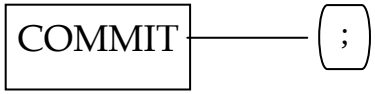
Sirve para retirar los privilegios otorgados. Al igual que ocurre con GRANT, tiene muchas opciones que no están contempladas en este pequeño resumen.



COMANDOS PARA EL CONTROL DE TRANSACCIONES. (Va asociado al concepto de transacción, en clase de teoría)

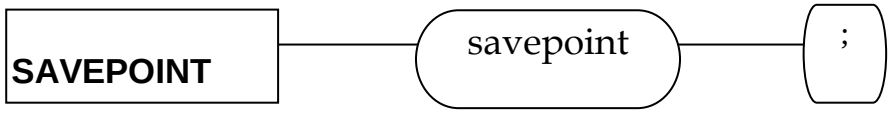
COMMIT

Cuando estamos ejecutando sentencias contra la base de datos, éstas no se hacen efectivas hasta que no ejecutamos un COMMIT. Hay algunas sentencias que llevan COMMIT asociado, por ejemplo CREATE TABLE. También implica un COMMIT abandonar una sesión de trabajo.



SAVEPOINT

Para marcar un punto dentro de un conjunto de sentencias que se están ejecutando.



ROLLBACK

Para volver atrás, deshacer acciones que se han realizado. Si ponemos savepoint se retrocede hasta ese punto. Si no especificamos savepoint se deshace toda la transacción.