

OODB SQL

Jason Rupard

Regular Expressions:

digit [0-9]

letter [a-zA-Z]

id letter(letter|digit|"_")*

sid letter(letter|digit|"_")*"." letter(letter|digit|"_")*

num "-"? (digit)+

real "-"? (digit)+ "." (digit)*

allowable_literal_symbol

"_ | "=" | "*" | "<" | "=" | ">" | "~" | "," | "(" | ")" | "[" | "]" | "." | "!" | "@" | "#" | "\$" | "%" | "^" | "&" | "+" | "?" | "(" | ")" | ":" | "|" | "-"

literal

"" (digit|letter|whitespace|allowable_literal_symbol)* ""

date

"" (digit)(digit)"/"(digit)(digit)"/"(digit)(digit)(digit)(digit) ""

Example Tables: (used for examples of statements)

Table::PRDCT

PNO	PNAME	COST	AVLQTY
P1	Modem	350	100
P2	Monitor	400	45
P3	Printer	700	15
P4	CPU Cards	2500	10
P5	Disk Unit	700	25
P6	Tape Drive	1200	27

Table::S

SNO	SNAME	QUOTA	CITY
S1	Job	4000	Dallas
S2	Baker	20000	Chicago
S3	Kirby	6000	Phoenix
S4	Sims	3000	San Diego
S5	Jones	30000	New York

Table::CUST

CNO	CNAME	CITY
C1	Sugar and Spice, CPA	San Diego
C2	ABC Product	New York
C3	Nachtfliegen, Inc.	Philadelphia
C4	Systems Group	Chicago
C5	JTA Inc.	Norman

Table::ORDERS

SNO	PNO	CNO	TTLQTY
S1	P1	C5	4
S1	P3	C5	2
S1	P3	C1	1
S2	P1	C1	2
S2	P4	C4	5
S2	P2	C4	5
S2	P3	C4	5
S3	P5	C4	5
S3	P6	C4	2
S5	P2	C2	10
S5	P4	C2	10
S5	P5	C2	10
S5	P6	C2	2
S5	P5	C3	3
S5	P3	C3	3

Statements:

ALTER:

- ALTER TABLE id ADD(id col_type[NOT NULL][UNIQUE],...);
 - Adds a column to table *id*.
 - col_type: DATE INT CHAR REAL.
 - **Example:**

```
ALTER TABLE ORDERS ADD (DAY DATE);
```

SNO	PNO	CNO	TTLQTY	DAY
S1	P1	C5	4	
S1	P3	C5	2	
S1	P3	C1	1	

...

- ALTER TABLE id DROP col_id;
 - Drops column *col_id* from table *id*.
 - **Example:**

```
ALTER TABLE ORDERS DROP DAY;
```

SNO	PNO	CNO	TTLQTY
S1	P1	C5	4
S1	P3	C5	2
S1	P3	C1	1

...

COMMIT:

- COMMIT;
 - Saves the current DB in memory to disk. The location of this DB is in `./oodb_dbs`.

CREATE:

- CREATE DATABASE id;
 - Creates a database with name *id*. If this DB is committed, then it saves as `./oodb_dbs/id.oodb`.
 - **Example:**

```
CREATE DATABASE Ranks;
```

- CREATE TABLE *id* (col_id col_type[NOT NULL][UNIQUE],...);
 - Creates a table with name *id*.
 - col_type: DATE INT CHAR REAL.

- **Example:**

```
CREATE TABLE CUST (
    CNO                                CHAR NOT NULL,
    CNAME                              CHAR,
    CITY                               CHAR);
```

```
+-----+-----+-----+
| CNO | CNAME | CITY |
+-----+-----+-----+
+-----+-----+-----+
```

- CREATE IMAGE "filename" AS *id*;
 - Creates a image from location of "*filename*" and stores it as *id* in the Catalog.
 - Image must be in PNG format.
 - Copies "*filename*" from its location can puts it in the "./oodb_dbs/" folder.

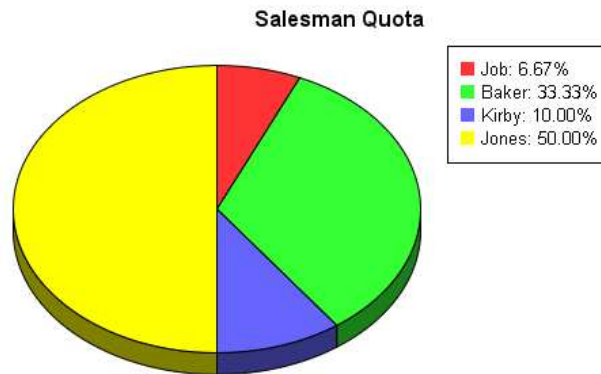
- **Example:**

```
CREATE IMAGE ".\my_image";
```

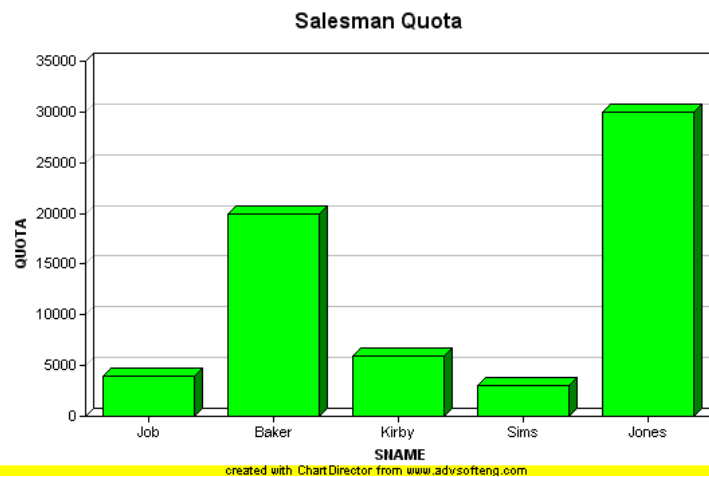
- Create [Bar|Pie] GRAPH *id* FROM *table_id* (attribute_name1, attribute_name2) [TITLE=literal] [WHERE predicate];
 - Creates a graph from *table_id*, stores it as *id* in the Catalog.
 - Either *attribute_name1* or *attribute_name2* (columns of *table_id*) must be of type INT or REAL.
 - *Title* will appear at the top of the graph.

- **Examples:**

```
CREATE PIE GRAPH S_GRAPH FROM S (SNAME,QUOTA)
TITLE='Salesman Quota'
WHERE QUOTA > 3000
;
```

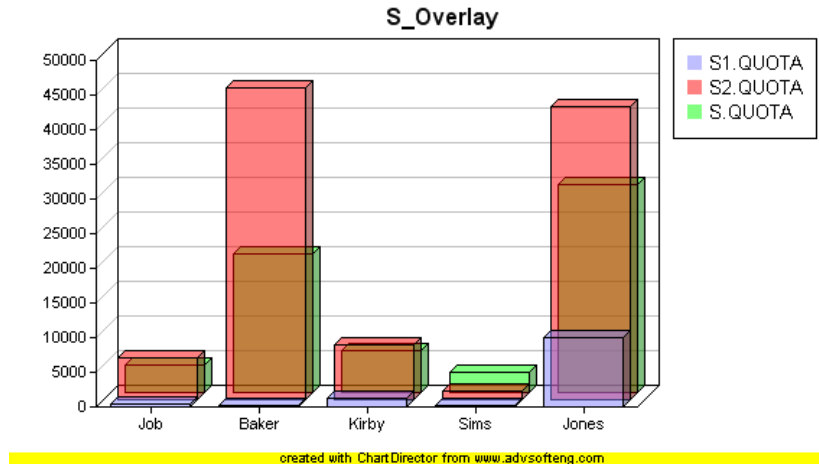


```
CREATE BAR GRAPH S_GRAPH FROM S (SNAME,QUOTA)
TITLE='Salesman Quota' ;
```



- CREATE OVERLAY GRAPH id FROM table_id (data_attribute_name)[,...] WITH LABELS sid [TITLE=literal];
 - o sid = table_id.column_id
 - o Max of 5 data sets for overlay command.
 - **Example:**

```
CREATE OVERLAY GRAPH S_Overlay FROM
S1(QUOTA),S2(QUOTA),S(QUOTA) WITH LABELS
S.SNAME;
```



DELETE:

- DELETE FROM id [WHERE where_expr];
 - Delete tuples from table *id*. If no WHERE clause is given, all tuples will be deleted.
 - **Example:**

```
DELETE FROM S WHERE !(SNO = 'S1' OR SNO = 'S2');
```

SNO	SNAME	QUOTA	CITY
S1	Job	4000	Dallas
S2	Baker	20000	Chicago

DROP:

- DROP object id;
 - Object → [TABLE|GRAPH|IMAGE]
 - Drops object *id* from current DB.
 - **Example:**

```
DROP IMAGE EYE;
```

INSERT:

- INSERT INTO id [(col_list)] VALUES (val[,val]...);
 - Inserts a tuple into table *id*.
 - If no *col_list* is given, the VALUES() list must specify one values of each column in the table. If a *col_list* is given consisting of one or more

comma-separated column names, one value per column must be specified in the VALUES() list.

- **Example:**

```
INSERT INTO PRDCT VALUES ('P1', 'Modem',  
350, 100);
```

```
+-----+-----+-----+-----+  
| PNO | PNAME | COST | AVLQTY |  
+-----+-----+-----+-----+  
| P1  | Modem | 350  | 100    |  
+-----+-----+-----+-----+
```

SELECT:

- SELECT [ALL|DISTINCT] col_sid,...|* FROM obj_list [WHERE where_expr];
 - o Retrieves rows from table if obj is a table.
 - o To show an Image: obj→ IMAGE(id)
 - Where id is the id of the image.
 - o To show a Graph that has already been created: GRAPH(id).
 - o col_sid: table_name.column_name is allowed.
 - o WHERE conditions can have the following operators
 - ! <> = <= >= < > ()
 - AND OR (as many as you want)
 - + - / * Arithmetic operations not implemented.
 - Possible to select (images|graphs) and join tables simultaneously.
 - **Example1:**

```
SELECT * FROM IMAGE(EYE), S, PRDCT, GRAPH(S_Q)  
WHERE !(S.SNO = 'S1') ;
```

Results, too much to paste.

- **Example2:**

```
SELECT DISTINCT CUST.CNO,CUST.CITY, S.CITY, S.SNO FROM CUST, S, ORDERS  
WHERE CUST.CITY <> S.CITY AND  
CUST.CNO = ORDERS.CNO AND S.SNO = ORDERS.SNO;
```

```
+-----+-----+-----+-----+  
| CUST.CNO | CUST.CITY | S.CITY | S.SNO |  
+-----+-----+-----+-----+  
| C5       | Norman   | Dallas | S1     |  
| C5       | Norman   | Dallas | S1     |
```

C1	San Diego	Dallas	S1
C1	San Diego	Chicago	S2
C4	Chicago	Phoenix	S3
C4	Chicago	Phoenix	S3
C3	Philadelphia	New York	S5
C3	Philadelphia	New York	S5

8 rows in set

- `SELECT [Bar|Pie] GRAPH id FROM table_id (attribute_name1, attribute_name2)[WHERE predicate] [TITLE=literal]`
 - Work just like `CREATE GRAPH` statement except it does not save the graph to the catalog. Once you close the graph's display window it is deleted.
- `SELECT OVERLAY GRAPH id FROM table_id (data_attribute_name)[,...] WITH LABELS sid [TITLE=literal];`
 - Work just like `CREATE OVERLAY` statement except it does not save the graph to the catalog. Once you close the graph's display window it is deleted.

SHOW:

- `SHOW object ID;`
 - Object `[TABLE|IMAGE|GRAPH]`
 - **Example1:**

```
SHOW TABLE PRDCT;
```

PNO	PNAME	COST	AVLQTY
P1	Modem	350	100
P2	Monitor	400	45
P3	Printer	700	15
P4	CPU Cards	2500	10
P5	Disk Unit	700	25
P6	Tape Drive	1200	27

Example2:

```
SHOW GRAPH S_Q;
```




created with ChartDirector from www.advsofteng.com

START:

- `START filename;`
 - Runs a file that contains SQL DML and DDL.
 - Not in WINDOWS GUI version.

UPDATE:

- `UPDATE id SET col_id=val[,col_id=val]...[WHERE where_expr];`
 - Modifies the contents of existing rows in the table *id*. The rows to be modified are those selected by the WHERE clause. For those rows that are selected, each column named in the SET clause is set to the corresponding *val*.

- **Example:**

```
UPDATE CUST SET CITY='Jacksonville'
WHERE CNO='C5';
```

CNO	CNAME	CITY
C1	Sugar and Spice, CPA	San Diego
C2	ABC Product	New York
C3	Nachtfliegen, Inc.	Philadelphia
C4	Systems Group	Chicago
C5	JTA Inc.	Jacksonville

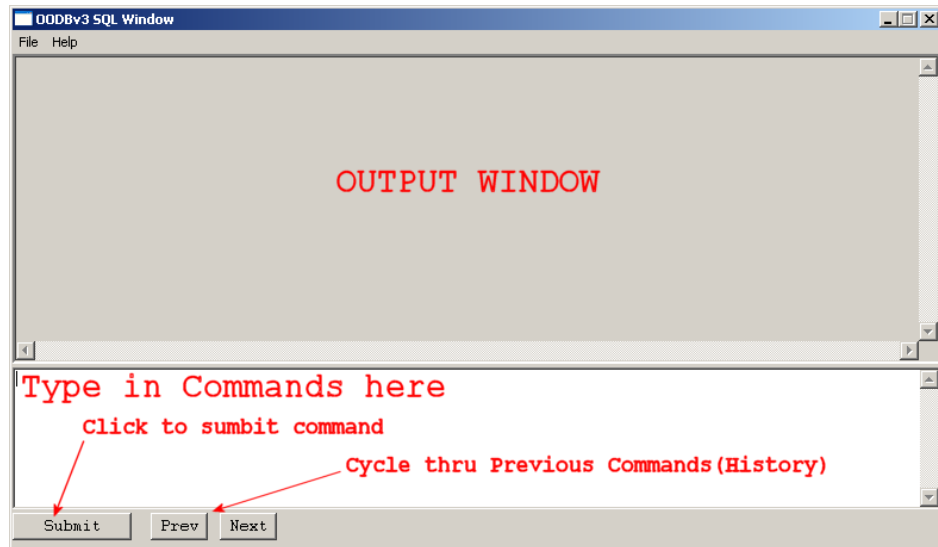
USE:

- `USE db_name;`
 - Select *db_name* to make it the current database.

- This command only looks for the database in the `./oodb_dbs/` directory.

Using OODB

- `oodbv3.exe`, `oodb_dbs/`, and `chartdir30.dll` must all be in the same directory.
 - If your running OODB.exe from MSVC++ refer to the “Compiling and Buliding” doc on how the files will be placed.



- Once OODB is up and running, the first thing is to load or create a database. You cannot do anything till you do this step:
 - `USE jaguar;`
 - Which loads `.\oodb_dbs\jaguar.oodb`
 - `CREATE DATABASE foobar;`
 - Which create a database for sratch → `.\oodb_dbs\foobar.oodb`
- Now that a database is loaded, OODB SQL DDL or DML can begin. Instructions on how to use OODB SQL commands are under the commands above.
- To save the changes made to a DB type:
 - `COMMIT;`
 - If you exit OODB without using `COMMIT` before hand, nothing will be save. ONLY when a `COMMIT` is made will data be saved.
- Anytime during the execution of OODB a new database can be brought in with the `USE` command:
 - `USE another_db;`
 - This will bring `another_db` into memory.
 - ONLY one database can be in memory at a time.

- If USE fails to load another_db. The previous database will still be in memory.
- Remember: The COMMIT command still has to be used to save a database. The USE command WILL NOT automatically save the work.
- Use the X button in the top right to exit OODB.