SQL for Oracle NoSQL Database Cheat Sheet

EXAMPLE

```
create table if not exists Example (
  id integer,
  firstname string,
  lastname string,
  age integer,
  income integer,
  address record(street string,
              city string,
              state string,
              phones array(record(type enum(work, home),
                                   areacode integer,
                                   number integer
                                  )
                          )
              ),
  connections array(integer),
  properties map(string),
  expenses map(integer),
  primary key(id)
)
```

```
{
  "id" : 1,
  "firstname" : "David",
  "lastname"  "Morrison",
  "age" : 25,:
  "income" : 100000,
  "address" : {"street":"150 Route 2",
           "city" : "Antioch",
           "state" : "TN",
           "phones":[{"type":"home", "areacode" : 423,
                     "number" : 8634379}]
           },
  "connections" : [2,3],
  "properties" : {"height" : "5.5", "weight" : "180"},
  "expenses" : {"books" : 500, "food" : 1000}
}

{
  "id" : 2,
  "firstname" : "John",
  "lastname" : "Anderson",
  "age" : 35,
  "income" : 100000,
  "address" : {"street":"187 Hill Street",
           "city" : "Beloit",
           "state" : "WI",
           "phones":[{"type":"home", "areacode" : 339,
                     "number" : 1684972}]
           },
  "connections" : [1,3],
  "properties" : {"height" : "6.0", "weight" : "190"},
  "expenses" : {"books" : 100, "food" : 800,
                "travel" : 10000}
}
```

QUERIES

ARITHMETIC OPERATORS	SELECT id, income, income/12 AS monthlysalary FROM Example;
ARRAY CONSTRUCTOR	SELECT lastName, [$e.address.phones[$element.areaCode = 423].number] AS phoneNumbers FROM Example $e;
COMPARISON OPERATORS	SELECT lastname FROM Example e WHERE e.address.state = "TN";
FIELDSTEP EXPRESSION	SELECT id, e.address.city FROM Example e WHERE e.address.state = "TN";
FILTERSTEP EXPRESSION	SELECT lastName FROM Example e WHERE e.address.phones[].areaCode =any 423;
FROM AS TABLE ALIAS	SELECT lastname FROM Example AS e;
FROM TABLE ALIAS	SELECT lastname FROM Example e;
FUNCTION CALL	SELECT id, size($e.address.phones) AS registeredphones FROM Example $e;

INDEX HINT	create index idx1 on Example (income);
	SELECT /*+ FORCE_INDEX(Example indx1) */* FROM Example where 90000 < income and income < 200000;

LOGICAL OPERATORS	SELECT lastname, age, income FROM Example WHERE age > 30 or income >= 100000;
ORDER BY ASC	SELECT id, lastname FROM Example ORDER BY id ASC;
ORDER BY DESC	SELECT id, lastname FROM Example ORDER BY id DESC;

ORDER BY INDEX	create index idx2 on Example (lastname);
	SELECT id, lastname FROM Example ORDER BY lastname;

ORDER BY PRIMARY KEY	SELECT id, lastname FROM Example ORDER BY id;
PARENTHESIZED EXPRESSION	SELECT id, lastname FROM Example WHERE (age > 20 or age < 40) and income >= 100000;
SELECT *	SELECT * FROM Example;
SELECT COLUMN(S)	SELECT firstname, lastname, age FROM Example;
SELECT COLUMN(S) AS	SELECT lastname AS Surname FROM Example;
SEQUENCE OPERATORS	SELECT id, lastname, connections FROM Example WHERE connections[] =any 2;
SLICESTEP EXPRESSIONS	SELECT [connections[0:1]] as strongConnections FROM Example WHERE id = 1;
WHERE	SELECT id, lastname FROM Example WHERE firstname = "John";

MAP FILTER STEPS	SELECT id, e.expenses.keys($value > 700) from Example e;
	SELECT id, e.expenses.keys($value > $.books) from Example e;
	SELECT id from Example e WHERE e.expenses.values($key != "books") >any 900;

SEARCHED CASE	SELECT id, CASE WHEN NOT EXISTS e.expenses.travel THEN "No Travel Expenses" ELSE e.expenses.travel end FROM Example e;

FUNCTIONS

size(item)	Returns the size of a complex Item (array, map, record).

OPERATORS

Arithmetic	+, -, *, /
Comparison	=, !=, >, >=, <, <=
Logical	AND, NOT, OR
Sequence	=any, !=any, >any, >=any, <=any
exists	Returns true if a sequence is not empty.

MAP FILTERS

.values(<expr>?)	Selects map field values.
.keys(<expr>?)	Selects map field keys.
$key	References key of the current field.
$value	References value of the current field.
$	References the entire map.

ARRAY FILTERS

[<expr>?]	Selects array elements.
$element	References current elements.
$pos	References position of current element.
$	References the entire array.

ARRAY SLICING

[<expr>? : <expr>?]	Selects array elements between two positions.
$	References the entire array.

CONSTRUCTORS

[<expr>*]	Array constructor
{(<expr> : <expr>)*}	Map constructor

SEARCHED CASE

```
CASE
    WHEN <expr> THEN <expr>
    (WHEN <expr> THEN <expr>)*
    (ELSE <expr>)?
END
```