SQL For Oracle NoSQL Database CheatSheet

EXAMPLE

```
create table 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),
 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"}
};
{
 "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"}
}
```

QUERIES

ARITHMETIC OPERATORS	SELECT id, income, income**/**12 AS monthlysalary FROM Example;
ARRAY CONSTRUCTOR	SELECT lastName, **[** $e.address.phone[$element.areaCode = 423].number **]** AS phoneNumbers FROM Example $e;
COMPARISON OPERATORS	SELECT lastname FROM Example WHERE age **>=** 25;
FIELDSTEP EXPRESSION	SELECT id, **e.address.city** FROM Example e WHERE **e.address.state = "TN";**
FILTERSTEP EXPRESSION	SELECT lastName FROM Example e WHERE **e.address.phone[].areaCode =any 423**;
FROM AS TABLE ALIAS	SELECT lastname **FROM** Example **AS e**;
FROM TABLE ALIAS	SELECT lastname **FROM** Example **e**;
FUNCTION CALL	SELECT id, **size($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 EXPRESSION	SELECT **[connections[0:1]]** as strongConnections FROM Example WHERE id = 1;
WHERE	SELECT id, lastname FROM Example **WHERE** firstname = "John";

FUNCTIONS

keys(item)	Returns the keys of a given record or map.
size(item)	Returns the size of a complex item (array, map, record).

OPERATORS

Arithmetic	+, -, *, /
Comparison	=, !=, >, >=, <, <=
Logical	AND, OR
Sequence	=any, !=any, >any, >=any, <any, <=any