UHD
UNIVERSITY OF HUMAN DEVELOPMENT

DATABASE MANAGEMENT SYSTEMS

Associate Professor Dr. Raed Ibraheem Hamed

University of Human Development, College of Science and Technology
Departments of IT and Computer Science

2015 – 2016

The ALTER TABLE Statement

The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.

SQL ALTER TABLE Syntax

To add a column in a table, use the following syntax:

```
ALTER TABLE table_name
ADD column_name datatype
```

The ALTER TABLE Statement

To delete a column in a table, use the following syntax:

```
ALTER TABLE table_name
DROP COLUMN column_name
```

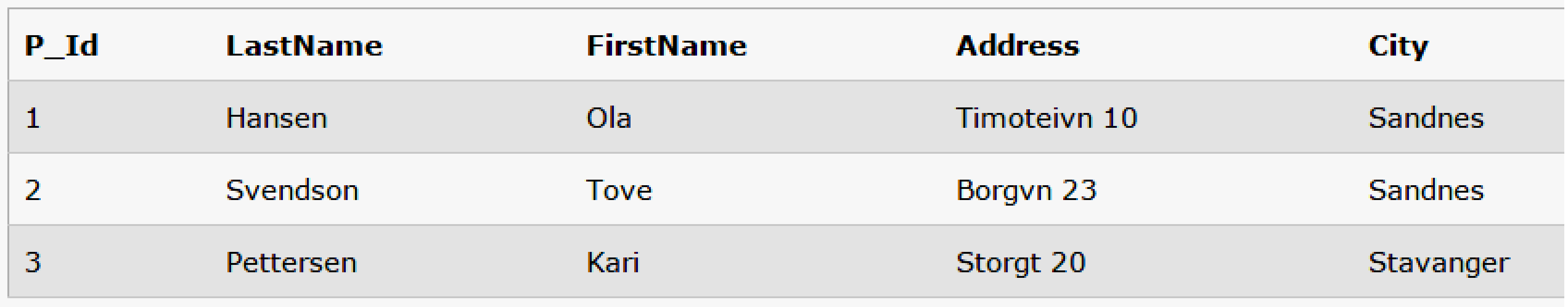
The ALTER TABLE Statement

To change the data type of a column in a table, use the following syntax:

```
ALTER TABLE table_name
ALTER  COLUMN column_name datatype
```

SQL ALTER TABLE Example

Look at the "Persons" table:



P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

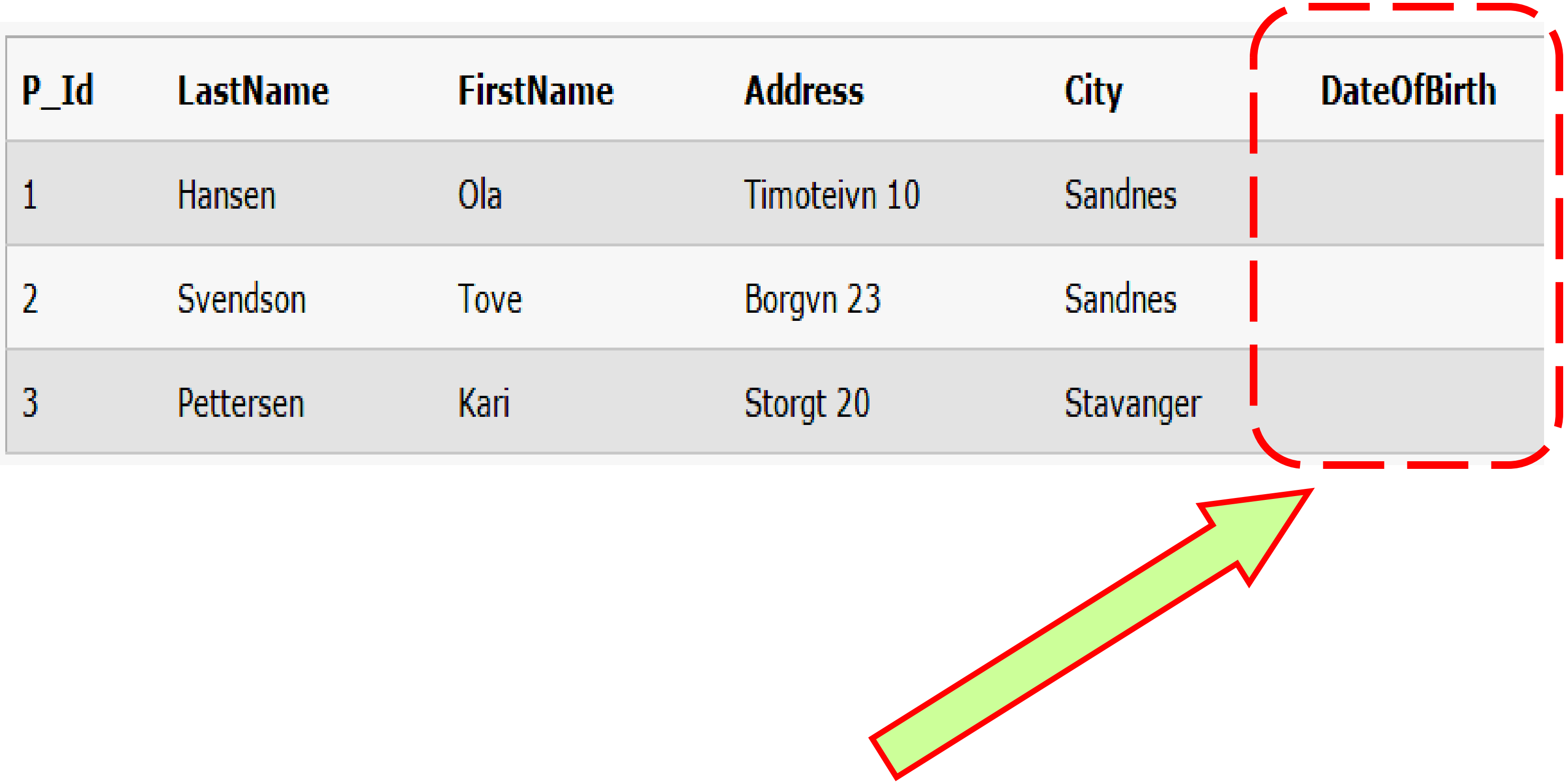
Now we want to add a column named "DateOfBirth" in the "Persons" table.

We use the following SQL statement:

```
ALTER TABLE Persons
ADD DateOfBirth date
```

SQL ALTER TABLE Example

The "Persons" table will now look like this:



P_Id	LastName	FirstName	Address	City	DateOfBirth
1	Hansen	Ola	Timoteivn 10	Sandnes	
2	Svendson	Tove	Borgvn 23	Sandnes	
3	Pettersen	Kari	Storgt 20	Stavanger	

Change Data Type Example

Now we want to change the data type of the column named "DateOfBirth" in the "Persons" table.

We use the following SQL statement:

```
ALTER TABLE Persons
ALTER COLUMN DateOfBirth  year
```

Notice that the "DateOfBirth" column is now of type **year** and is going to hold a year in a **two-digit or four-digit** format.

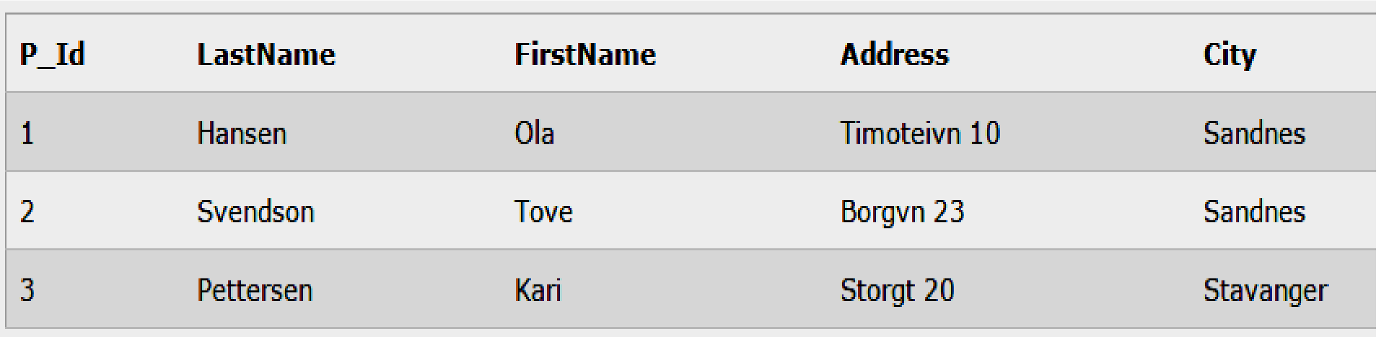
DROP COLUMN Example

Next, we want to delete the column named "DateOfBirth" in the "Persons" table.

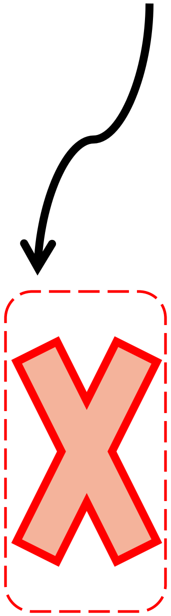
We use the following SQL statement:

```
ALTER TABLE Persons
DROP COLUMN DateOfBirth
```

The "Persons" table will now look like this:



P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger



SQL AUTO INCREMENT Field

Auto-increment allows a unique number to be generated when a new record is inserted into a table.

AUTO INCREMENT a Field

Very often we would like the value of the primary key field to be created automatically every time a new record is inserted.

We would like to create an auto-increment field in a table.

MySQL uses the **AUTO_INCREMENT** keyword to perform an auto-increment feature.

SQL AUTO INCREMENT Field

Syntax for MySQL

The following SQL statement defines the "ID" column to be an auto-increment primary key field in the "**Persons**" table:

```
CREATE TABLE Persons
(
ID int NOT NULL AUTO_INCREMENT,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Address varchar(255),
City varchar(255),
PRIMARY KEY (ID)
)
```

SQL AUTO INCREMENT Field

By default, the starting value for AUTO_INCREMENT is 1, and it will increment by 1 for each new record.

To let the AUTO_INCREMENT sequence **start with another value**, use the following SQL statement:

```
ALTER TABLE Persons AUTO_INCREMENT=100
```

To insert a new record into the "Persons" table, we will NOT have to specify a value for the "ID" column (a unique value will be added automatically):

```
INSERT INTO Persons (FirstName, LastName)
VALUES ('Lara', ‘Rock')
```

The "ID" column would be assigned a unique value. The "FirstName" column would be set to "Lara" and the "LastName" column would be set to “Rock“.

SQL Dates

The most difficult part when working with dates is to **be sure that the format of the date** you are trying to **insert, matches** the format of the **date column in the database.**

Sunday, July 21, 2013

SQL Date Data Types

MySQL comes with the following data types for storing a date or a date/time value in the database:

- DATE - format YYYY-MM-DD
- DATETIME - format: YYYY-MM-DD HH:MI:SS
- TIMESTAMP - format: YYYY-MM-DD HH:MI:SS
- YEAR - format YYYY or YY

The supported range for DATETIME type is:

'1000-01-01 00:00:00' to '9999-12-31 23:59:59'.

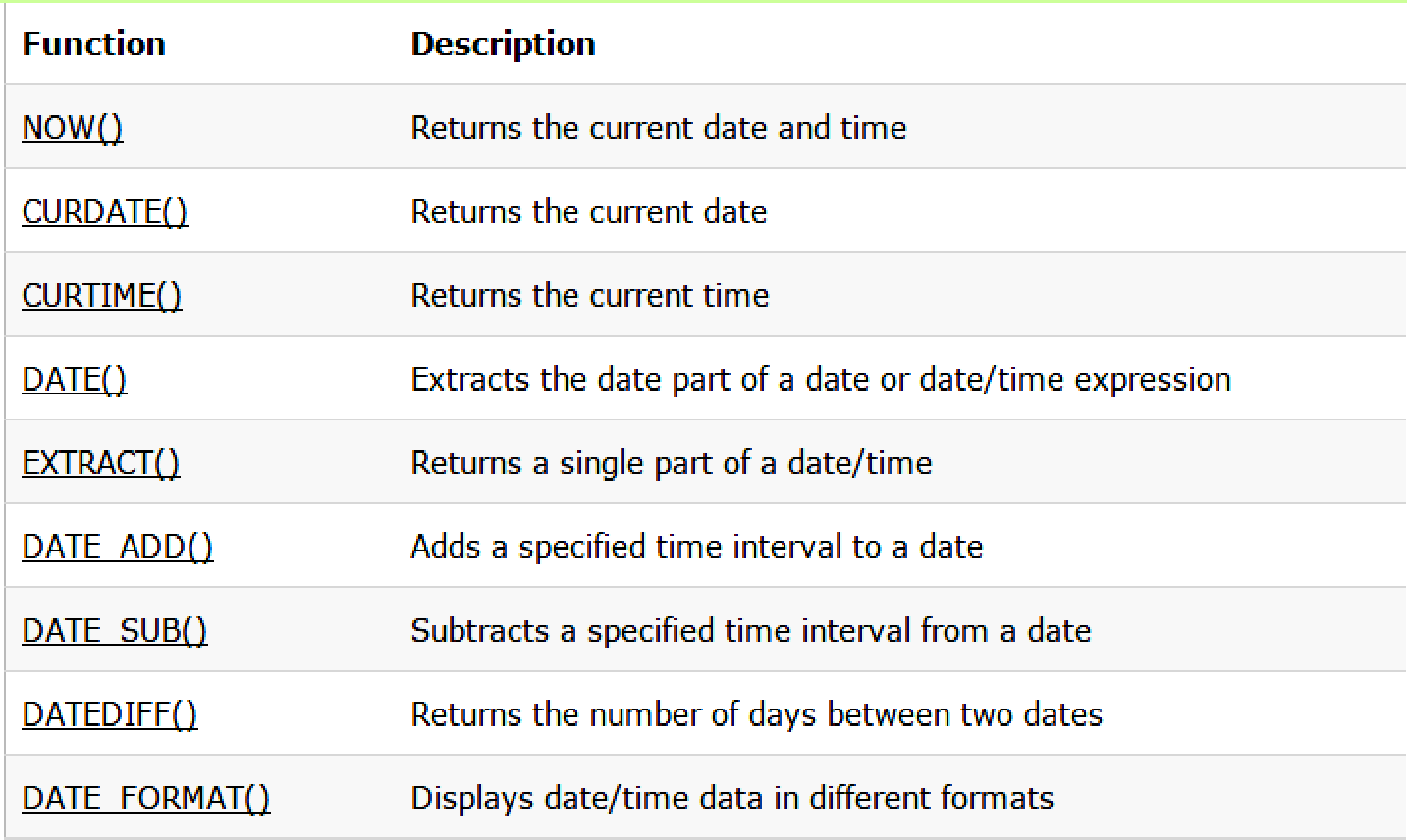
The supported range for TIMESTAMP type is:

'1970-01-01 00:00:01' to '2038-01-19 03:14:07'.

That means if you want to store date which is before the year 1970 or after the year 2038 you will need to use DATETIME.

MySQL Date Functions

Before talking about the complications of querying for dates, we will look at the most important built-in functions for working with dates.



Function	Description
NOW()	Returns the current date and time
CURDATE()	Returns the current date
CURTIME()	Returns the current time
DATE()	Extracts the date part of a date or date/time expression
EXTRACT()	Returns a single part of a date/time
DATE_ADD()	Adds a specified time interval to a date
DATE_SUB()	Subtracts a specified time interval from a date
DATEDIFF()	Returns the number of days between two dates
DATE_FORMAT()	Displays date/time data in different formats

MySQL Date Syntax

Syntax

NOW()

Example:-

The following SELECT statement:

```
SELECT NOW(),CURDATE(),CURTIME()
```

will result in something like this:

NOW()	CURDATE()	CURTIME()
2014-11-22 12:45:34	2014-11-22	12:45:34

MySQL Date Syntax

Example

The following SQL creates an "Orders" table with a datetime column (OrderDate):

```
CREATE TABLE Orders
(
OrderId int NOT NULL,
ProductName varchar(50) NOT NULL,
OrderDate datetime NOT NULL DEFAULT NOW(),
PRIMARY KEY (OrderId)
)
```

MySQL Date Syntax

Notice that the OrderDate column specifies NOW() as the default value. As a result, when you insert a row into the table, the current date and time are automatically inserted into the column.

Now we want to insert a record into the "Orders" table:

```
INSERT INTO Orders (ProductName) VALUES ('Jarlsberg Cheese')
```

The "Orders" table will now look something like this:

OrderId	ProductName	OrderDate
1	Jarlsberg Cheese	2014-11-22 13:23:44.657

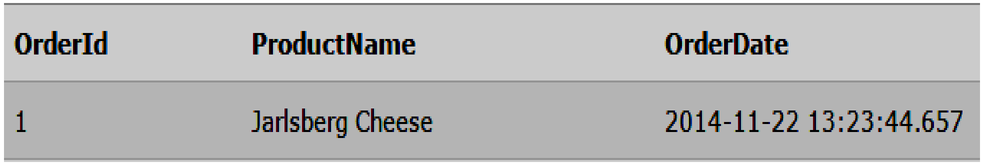
MySQL Date Syntax

Syntax

DATE(date)

The following SELECT statement:

```
SELECT ProductName, DATE(OrderDate) AS OrderDate
FROM Orders
WHERE OrderId=1
```



OrderId	ProductName	OrderDate
1	Jarlsberg Cheese	2014-11-22 13:23:44.657

will result in this:

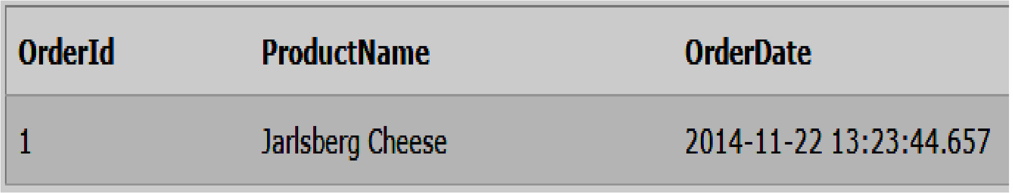
ProductName	OrderDate
Jarlsberg Cheese	2014-11-22

SQL Date Data Types

Syntax

EXTRACT(unit FROM date)

The EXTRACT() function is used to return a single part of a date/time, such as year, month, day, hour, minute, etc.

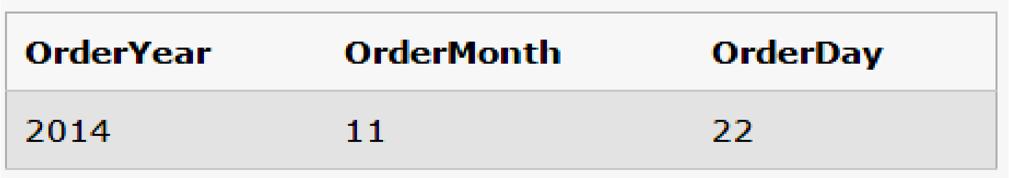


OrderId	ProductName	OrderDate
1	Jarlsberg Cheese	2014-11-22 13:23:44.657

The following SELECT statement:

```
SELECT EXTRACT(YEAR FROM OrderDate) AS OrderYear,
EXTRACT(MONTH FROM OrderDate) AS OrderMonth,
EXTRACT(DAY FROM OrderDate) AS OrderDay,
FROM Orders
WHERE OrderId=1
```

will result in this:



OrderYear	OrderMonth	OrderDay
2014	11	22

SQL Date Data Types

Syntax

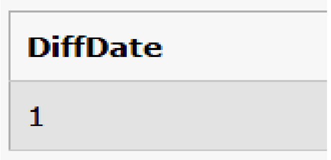
DATEDIFF(date1,date2)

Example

The following SELECT statement:

SELECT DATEDIFF('2014-11-30','2014-11-29') AS DiffDate

will result in this:



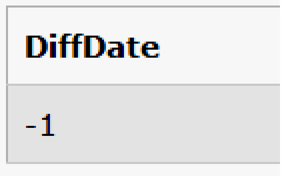
DiffDate
1

Example

The following SELECT statement:

SELECT DATEDIFF('2014-11-29','2014-11-30') AS DiffDate

will result in this:



DiffDate
-1

SQL NULL Values

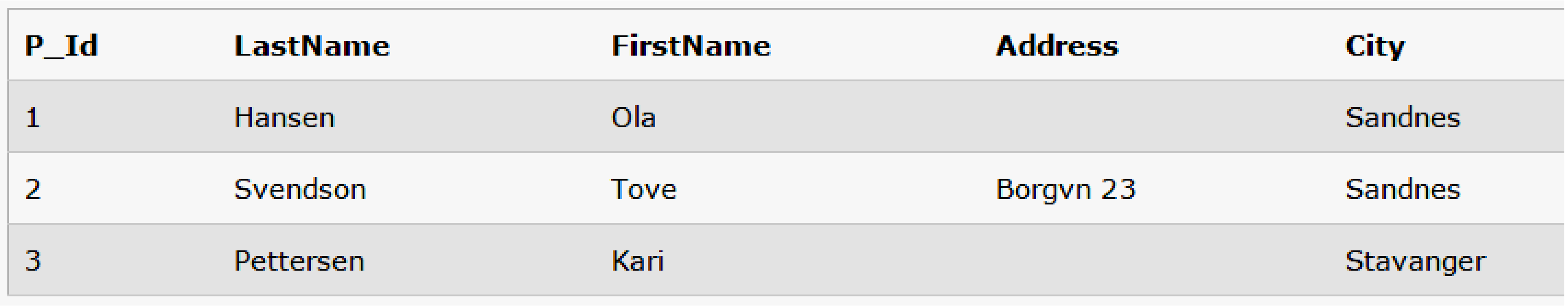
a) NULL values represent missing unknown data.

b) By default, a table column can hold NULL values.

c) We can insert a new record or update an existing record without adding a value to this column.

d) It is not possible to compare NULL and 0; they are not equivalent.

Look at the following "Persons" table:



P_Id	LastName	FirstName	Address	City
1	Hansen	Ola		Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari		Stavanger

SQL IS NULL

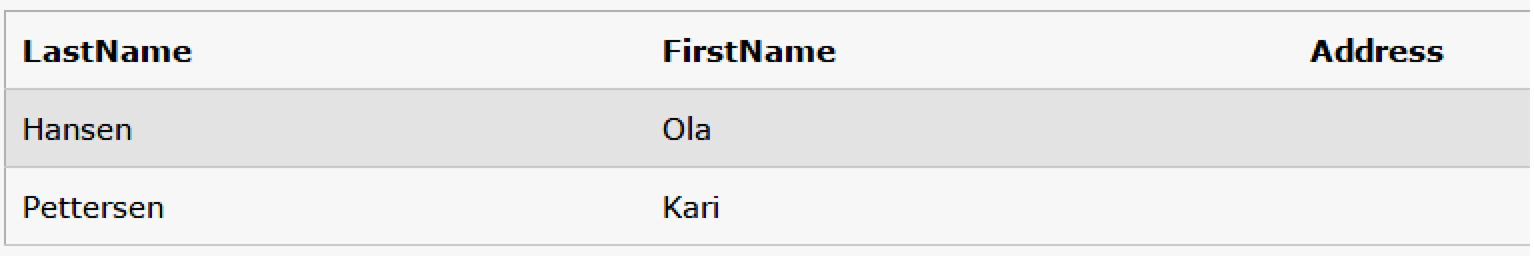
SQL IS NULL

How do we select only the records with NULL values in the "Address" column?

We will have to use the IS NULL operator:

```
SELECT LastName, FirstName, Address FROM Persons
WHERE Address IS NULL
```

The result-set will look like this:



LastName	FirstName	Address
Hansen	Ola	
Pettersen	Kari	

SQL IS NOT NULL

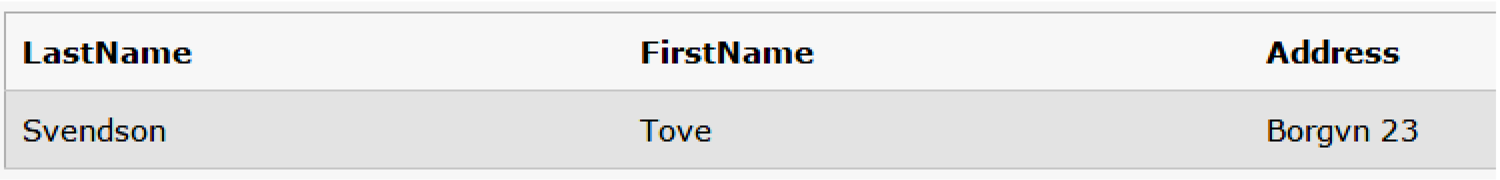
SQL IS NOT NULL

How do we select only the records with no NULL values in the "Address" column?

We will have to use the IS NOT NULL operator:

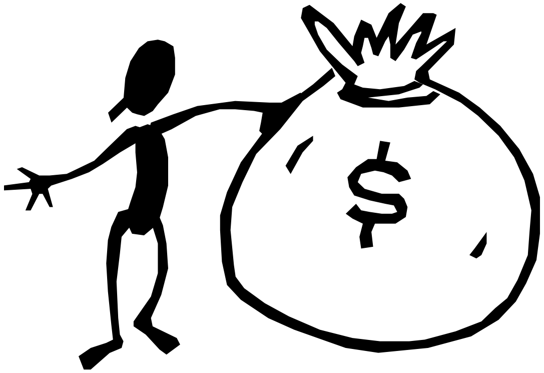
```
SELECT LastName, FirstName, Address FROM Persons
WHERE Address IS NOT NULL
```

The result-set will look like this:



LastName	FirstName	Address
Svendson	Tove	Borgvn 23

Thank you



???