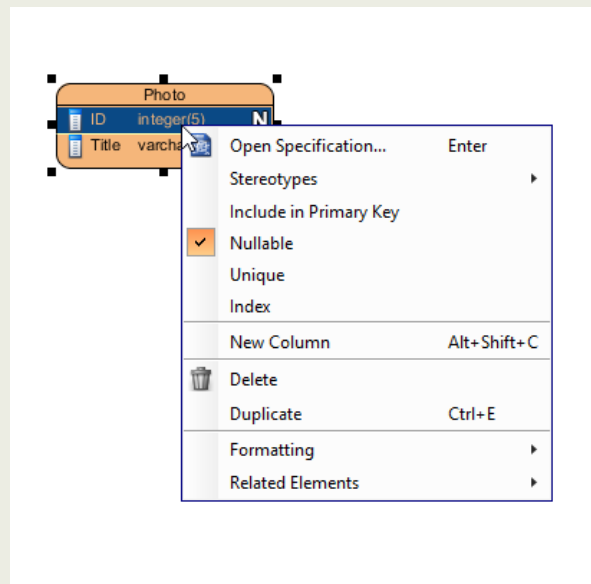


Column in Detail

Column in Detail

Opening column specification

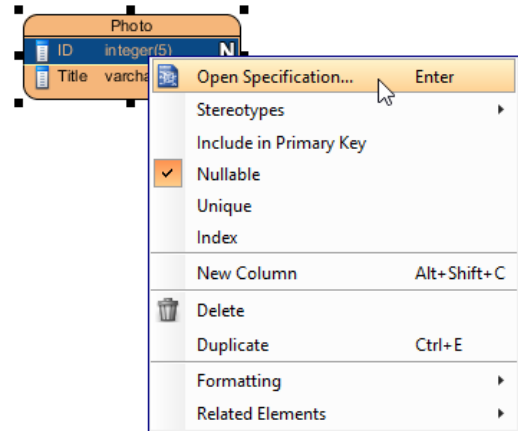
1. Right click on a column.



Column in Detail

Opening column specification

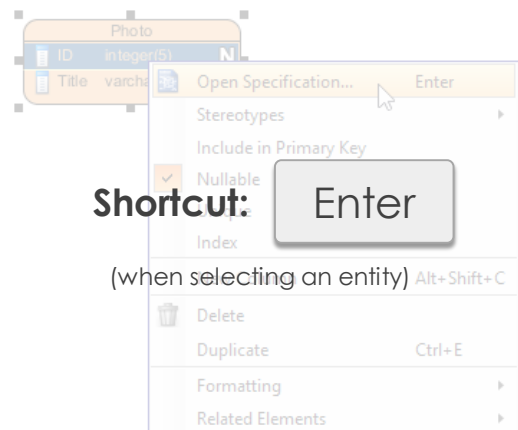
1. Right click on a column.
2. Select **Open Specification...** from the popup menu.



Column in Detail

Opening column specification

1. Right click on a column.
2. Select **Open Specification...** from the popup menu.



Column in Detail

Opening column specification

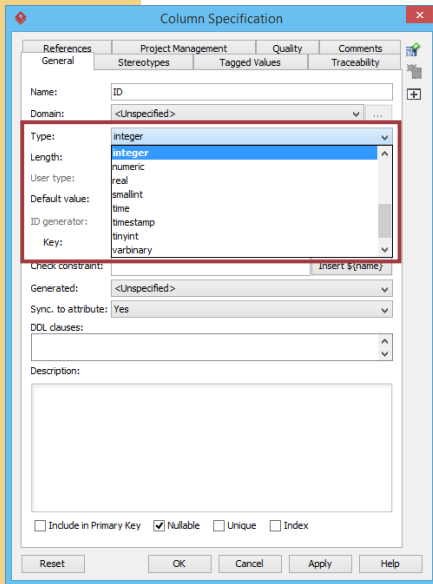
Column in Detail

Column details ■ Domain

- Definition: A named set of property values that can be re-used by a column.
- Usage
 1. Create a domain for commonly entered values such as type, length and default value.
 2. Re-use it in other columns when needed.
- Benefits
 - Save time in creating **similar** columns
 - Keep data model consistent

Column in Detail

Column details ■ Type

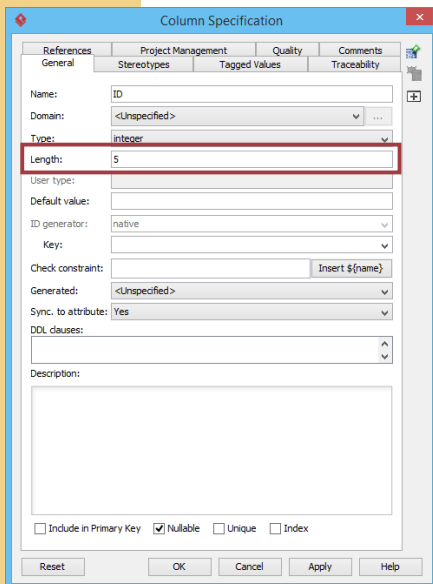


The screenshot shows the 'Column Specification' dialog box with the 'General' tab selected. The 'Type' dropdown is highlighted with a red box, showing a list of data types: integer, numeric, real, smallint, time, timestamp, binary, and varbinary. The 'Length' field is also highlighted with a red box, showing the value '5'.

- Identify the type of data allowed to be stored in the column.
 - E.g. integer, char, varchar, long, float, datetime
- Different vendors may support the same type differently
 - E.g.
 - decimal in MySQL, float in Oracle
 - timestamp in Oracle, datetime in MSSQL
- You can configure the DBMS you use in Database Configuration (**Tools > DB > Database Configuration**)

Column in Detail

Column details ■ Length



The screenshot shows the 'Column Specification' dialog box with the 'General' tab selected. The 'Length' field is highlighted with a red box, showing the value '5'.

- The maximum allowable length of data that can be stored under the column, per each record.

Column in Detail

Column details

User Type

The screenshot shows the 'Column Specification' dialog box with the 'General' tab selected. The 'User type' field is highlighted with a red rectangle and contains the text 'ENUM('Only You','You & Friends','Public')'. Other fields include Name: Privacy, Domain: <Unspecified>, Type: varchar, Length: 10, Default value: (empty), ID generator: native, Key: (empty), Check constraint: Insert \$(name), Generated: <Unspecified>, Sync. to attribute: Yes, and DDL clauses: (empty). At the bottom, there are checkboxes for 'Include in Primary Key', 'Nullable' (checked), 'Unique', and 'Index', along with buttons for 'Reset', 'OK', 'Cancel', 'Apply', and 'Help'.

- Define enumeration values
- Effective in database generation

Column in Detail

Column details

Default Value

The screenshot shows the 'Column Specification' dialog box with the 'General' tab selected. The 'Default value' field is highlighted with a red rectangle and contains the text 'Only You'. Other fields are the same as in the previous screenshot: Name: Privacy, Domain: <Unspecified>, Type: varchar, Length: 10, User type: ENUM('Only You','You & Friends','Public'), ID generator: native, Key: (empty), Check constraint: Insert \$(name), Generated: <Unspecified>, Sync. to attribute: Yes, and DDL clauses: (empty). At the bottom, there are checkboxes for 'Include in Primary Key', 'Nullable' (checked), 'Unique', and 'Index', along with buttons for 'Reset', 'OK', 'Cancel', 'Apply', and 'Help'.

- A default value of the column when no value is specified when inserting data into the database.

Column in Detail

Column details

ID Generator

Column Specification

References

General

Project Management

Quality

Comments

Stereotypes

Tagged Values

Traceability

Name:

ID

Domain:

<Unspecified>

Type:

int

Length:

5

User type:

Default values:

ID generator:

native

Key:

identity

Check constraint:

native

Generated:

sequence

Sync. to attribute:

uuid.hex

DDL clauses:

uuid.string

Description:

☒ Include in Primary Key

☐ Nullable

☐ Unique

☐ Index

Reset

OK

Cancel

Apply

Help

- The strategy of how unique values will be produced in a **primary key** column.

Column details

ID Generator

ID Generator	Description
assigned	allows the application to assign an identifier to the object before save() is called.
guid	uses a database-generated GUID string on MS SQL Server and MySQL.
hilo	uses a hi/lo algorithm to efficiently generate identifiers of type long, short or int, given a table and column as a source of hi values. The hi/lo algorithm generates identifiers that are unique only for a particular database.
identity	supports identity columns in DB2, MySQL, MS SQL Server, Sybase and HypersonicSQL. The returned identifier is of type long, short or int.
increment	generates identifiers of type long, short or int that are unique only when no other process is inserting data into the same table. Do not use in a cluster.
native	(default) picks identity, sequence or hilo depending upon the capabilities of the underlying database.
seqhilo	uses a hi/lo algorithm to efficiently generate identifiers of type long, short or int, given a named database sequence.
Sequence	uses a sequence in DB2, PostgreSQL, Oracle. The returned identifier is of type long,short or int

Column details

Check Constraint

Column Specification

References: General, Project Management, Quality, Comments
Stereotypes, Tagged Values, Traceability

Name: ID

Domain: <Unspecified>

Type: int

Length: 5

User type:

Default value:

ID generator: native

Key:

Check constraint: ID > 1000 Insert \$(name)

Generated: <Unspecified>

Sync. to attribute: Yes

DDL clauses:

Description:

☐ Include in Primary Key ☐ Nullable ☐ Unique ☐ Index

Reset OK Cancel Apply Help

- Allows only certain values to be filled in a column.

Column in Detail

Column details

Generated

Column Specification

References: General, Project Management, Quality, Comments
Stereotypes, Tagged Values, Traceability

Name: ID

Domain: <Unspecified>

Type: int

Length: 5

User type:

Default value:

ID generator: native

Key:

Check constraint: Insert \$(name)

Generated: <Unspecified>

Sync. to attribute: Never

DDL clauses: Always

Description:

☐ Include in Primary Key ☐ Nullable ☐ Unique ☐ Index

Reset OK Cancel Apply Help

- Used for ORM. It marks the column to be generated when insert, or both insert/update
- e.g. a trigger will update column "lastModified" by current time when insert or update. When generated = always, Hibernate will execute "select lastModified ..." after insert/update to update the ORM model.

Column in Detail

Column details

Sync. to Attribute

The screenshot shows the 'Column Specification' dialog box with the following fields and values:

- Name: ID
- Domain: <Unspecified>
- Type: int
- Length: 5
- User type:
- Default value:
- ID generator: native
- Key:
- Check constraint: Insert \$(name)
- Generated: <Unspecified>
- Sync. to attribute: Yes (highlighted with a red box)
- DDL clauses: Yes (highlighted with a red box)
- Description:
- Include in Primary Key: ☐ Nullable: ☐ Unique: ☐ Index: ☐

- Enable/disable synchronizing the column with the attribute in the corresponding class
 - Column name <-> Attribute name
 - Column type <-> Attribute data type

Column in Detail

Column details

DDL Clauses

The screenshot shows the 'Column Specification' dialog box with the following fields and values:

- Name: ID
- Domain: <Unspecified>
- Type: int
- Length: 5
- User type:
- Default value:
- ID generator: native
- Key:
- Check constraint: Insert \$(name)
- Generated: <Unspecified>
- Sync. to attribute: Yes
- DDL clauses: STORAGE DISK (highlighted with a red box)
- Description:
- Include in Primary Key: ☐ Nullable: ☐ Unique: ☐ Index: ☐

- The text to include in the create table statement in database generation

Column in Detail

Column details

Description

Column Specification

References General Project Management Quality Comments
Stereotypes Tagged Values Traceability

Name: ID
Domain: <Unspecified>
Type: int
Length: 5
User type:
Default value:
ID generator: native
Key:
Check constraint: Insert \$(name)
Generated: <Unspecified>
Sync. to attribute: Yes
DDL clauses:
Description:
Internal ID of photo. It is a unique and non-editable field. It's expected to begin from 1,000.
☐ Include in Primary Key ☐ Nullable ☐ Unique ☐ Index
Reset OK Cancel Apply Help

- Describe the column in detail
- Useful in:
 - Creating data dictionary
 - Documenting database table (as comment of columns)

Column in Detail

Column details

Include in Primary Key

Column Specification

References General Project Management Quality Comments
Stereotypes Tagged Values Traceability

Name: ID
Domain: <Unspecified>
Type: int
Length: 5
User type:
Default value:
ID generator: native
Key:
Check constraint: Insert \$(name)
Generated: <Unspecified>
Sync. to attribute: Yes
DDL clauses:
Description:
Internal ID of photo. It is a unique and non-editable field. It's expected to begin from 1,000.
☒ Include in Primary Key ☐ Nullable ☐ Unique ☐ Index
Reset OK Cancel Apply Help

- A primary key (PK) uniquely identifies each record in a table.
- One entity can have zero or one primary key.
- One primary key can be formed by multiple columns.

Column in Detail

Column details

Nullable

The screenshot shows the 'Column Specification' dialog box with the 'General' tab selected. The 'Name' field is 'ID', 'Domain' is '<Unspecified>', 'Type' is 'int', 'Length' is '5', 'User type' is empty, 'Default value' is empty, 'ID generator' is 'native', 'Key' is empty, 'Check constraint' is empty, 'Generated' is '<Unspecified>', 'Sync. to attribute' is 'Yes', and 'DDL clauses' is empty. The 'Description' field contains the text: 'Internal ID of photo. It is a unique and non-editable field. It's expected to begin from 1,000'. At the bottom, the 'Include in Primary Key' checkbox is unchecked, the 'Nullable' checkbox is checked (highlighted with a red box), the 'Unique' checkbox is unchecked, and the 'Index' checkbox is unchecked. The 'Reset', 'OK', 'Cancel', 'Apply', and 'Help' buttons are at the bottom.

- Allow value of the column to be set as NULL
- Situations when applicable:
 - Value is to be filled later
 - Value can be unspecified
- Make sure the use of nullable field does not break your business rule

Column in Detail

Column details

Unique

The screenshot shows the 'Column Specification' dialog box with the 'General' tab selected. The 'Name' field is 'ID', 'Domain' is '<Unspecified>', 'Type' is 'int', 'Length' is '5', 'User type' is empty, 'Default value' is empty, 'ID generator' is 'native', 'Key' is empty, 'Check constraint' is empty, 'Generated' is '<Unspecified>', 'Sync. to attribute' is 'Yes', and 'DDL clauses' is empty. The 'Description' field contains the text: 'Internal ID of photo. It is a unique and non-editable field. It's expected to begin from 1,000'. At the bottom, the 'Include in Primary Key' checkbox is unchecked, the 'Nullable' checkbox is unchecked, the 'Unique' checkbox is checked (highlighted with a red box), and the 'Index' checkbox is unchecked. The 'Reset', 'OK', 'Cancel', 'Apply', and 'Help' buttons are at the bottom.

- Used to ensure data is not duplicated in any two rows in a database table.
- Can be null

Column in Detail

Column details

Index

Column Specification

References: General, Project Management, Quality, Comments
Stereotypes, Tagged Values, Traceability

Name: ID
Domain: <Unspecified>
Type: int
Length: 5
User type:
Default value:
ID generator: native
Key:
Check constraint: Insert \$(name)
Generated: <Unspecified>
Sync. to attribute: Yes
DDL clauses:
Description:
Internal ID of photo. It is a unique and non-editable field. It's expected to begin from 1,000
☐ Include in Primary Key ☐ Nullable ☐ Unique ☒ Index
Reset OK Cancel Apply Help

- Index improves the speed of data retrieval on a table.
- You set a column to be the index column when data retrieval has to be made frequently through that column.

Column in Detail

Index (Entity)







Column in Detail

Querying data without index

Uploader Email	Uploader Name	Uploader Address
donald.t.lopresti@gmail.com	Donald T. Lopresti	572 Lost Lake Crossing
michele.r.herrmann@gmail.com	Michele R. Herrmann	213 White Swan Garden
wa		asse
car		reet
kin		ue
Let's find him!  tony.stanley.ca@gmail.com		
⋮		
tony.stanley.ca@gmail.com	Tony Stanley	2823 Percy Avenue
heleb.b.broom@gmail.com	Helen Broom	746 Happy Valley

Column in Detail

Querying data without index

	Uploader Email	Uploader Name	Uploader Address
	donald.t.lopresti@gmail.com	Donald T. Lopresti	572 Lost Lake Crossing
	michele.r.herrmann@gmail.com	Michele R. Herrmann	213 White Swan Garden
	wanda.a.Allison@gmail.com	Wanda A. Allison	2017 Green Impasse
	carol.faulkner.us@gmail.com	Carol Faulkner	2841 Rodney Street
	kimkim.g.1980@gmail.com	Kimberly Greenlee	2314 Pearl Avenue
■	⋮		
■			
■			
	tony.stanley.ca@gmail.com	Tony Stanley	2823 Percy Avenue
	heleb.b.broom@gmail.com	Helen Broom	746 Happy Valley

Column in Detail

Index speeds up search queries by
**reducing the number of records that
need to be examined**

Querying data

Index Column

Uploader Email	Uploader Name	Uploader Address
donald.f.lopresti@gmail.com	Donald T. Lopresti	572 Lost Lake Crossing
michele.r.herrmann@gmail.com	Michele R. Herrmann	213 White Swan Garden
wanda.a.Allison@gmail.com	Wanda A. Allison	2017 Green Impasse
carol.faulkner.us@gmail.com	Carol Faulkner	2841 Rodney Street
kimkim.g.1980@gmail.com	Kimberly Greenlee	2314 Pearl Avenue
⋮		
tony.stanley.ca@gmail.com	Tony Stanley	2823 Percy Avenue
heleb.b.broom@gmail.com	Helen Broom	746 Happy Valley

Querying data with index

Uploader Email	Uploader Name	Uploader Address
donald.t.lopresti@gmail.com	Donald T. Lopresti	572 Lost Lake Crossing
michele.r.herrmann@gmail.com	Michele R. Herrmann	213 White Swan Garden
wanda.a.Allison@gmail.com	Wanda A. Allison	2017 Green Impasse
carol.faulkner.us@gmail.com	Index Table	2841 Rodney Street
kimkim.g.1980@gmail.com		2314 Pearl Avenue
tony.stanley.ca@gmail.com	Tony Stanley	2823 Percy Avenue
heleb.b.broom@gmail.com	Helen Broom	746 Happy Valley

Column in Detail

Querying data with index



Uploader Email
donald.t.lopresti@gmail.com
michele.r.herrmann@gmail.com
wanda.a.Allison@gmail.com
carol.faulkner.us@gmail.com
kimkim.g.1980@gmail.com
⋮
tony.stanley.ca@gmail.com
heleb.b.broom@gmail.com

Column in Detail

Querying data with index



BINARY
SEARCH

Uploader Email
carol.faulkner.us@gmail.com
donald.t.lopresti@gmail.com
heleb.b.broom@gmail.com
kimkim.g.1980@gmail.com
michele.r.herrmann@gmail.com
⋮
tony.stanley.ca@gmail.com
wanda.a.Allison@gmail.com

Column in Detail

Querying data with index

kimkim.g.1980@gmail.com

VS

tony.stanley.ca@gmail.com



Uploader Email
carol.faulkner.us@gmail.com
donald.t.lopresti@gmail.com
heleb.b.broom@gmail.com
kimkim.g.1980@gmail.com
michele.r.herrmann@gmail.com
⋮
tony.stanley.ca@gmail.com
wanda.a.Allison@gmail.com

Column in Detail

Querying data with index

Uploader Email	Uploader Email	Uploader Name	Uploader Address
carol.faulkner.us@gmail.com	donald.t.lopresti@gmail.com	Donald T. Lopresti	572 Lost Lake Crossing
donald.t.lopresti@gmail.com	michele.r.herrmann@gmail.com	Michele R. Herrmann	213 White Swan Garden
heleb.b.broom@gmail.com	wanda.a.Allison@gmail.com	Wanda A. Allison	2017 Green Impasse
kimkim.g.1980@gmail.com	carol.faulkner.us@gmail.com	Carol Faulkner	2841 Rodney Street
michele.r.herrmann@gmail.com	kimkim.g.1980@gmail.com	Kimberly Greenlee	2314 Pearl Avenue
⋮	⋮		
tony.stanley.ca@gmail.com	tony.stanley.ca@gmail.com	Tony Stanley	2823 Percy Avenue
wanda.a.Allison@gmail.com	heleb.b.broom@gmail.com	Helen Broom	746 Happy Valley

Column in Detail

Considerations of Using Index

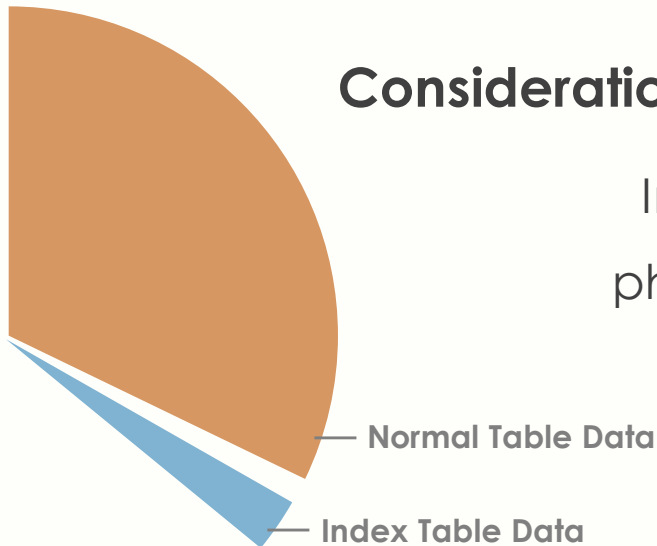
Index – Use or Not?

Column in Detail

Considerations of Using Index

Consideration 1: Storage space

Index table consumes physical storage space

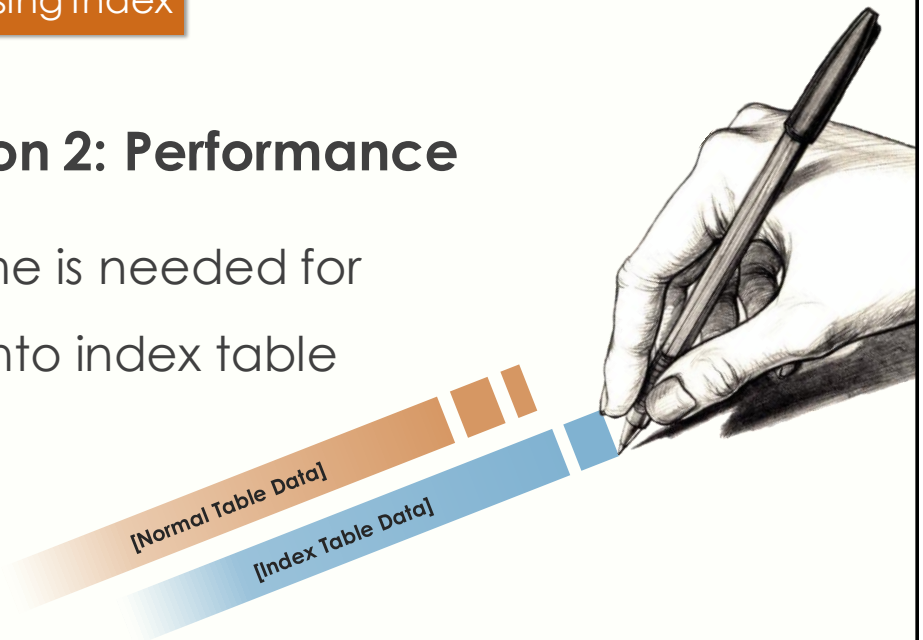


Column in Detail

Considerations of Using Index

Consideration 2: Performance

Additional time is needed for **writing data** into index table



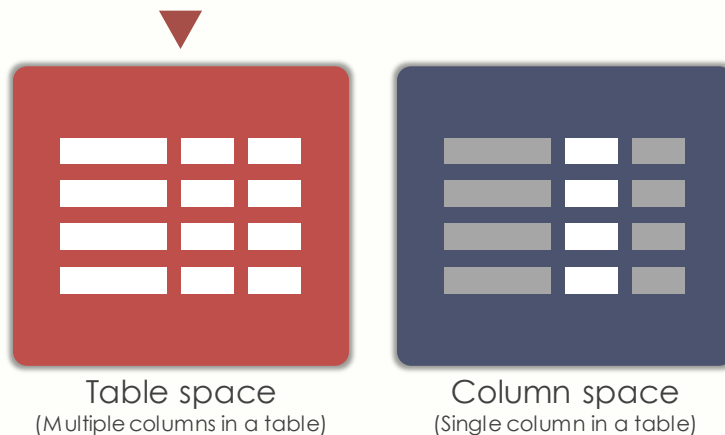
Column in Detail

Considerations of Using Index

Index is used when we have an often need of **reading** data than **writing** data.

Column in Detail

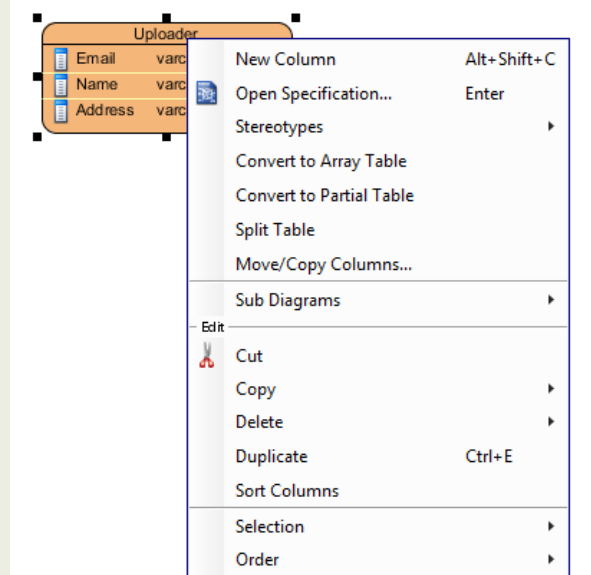
Defining index



Column in Detail

Defining index ■ Table space

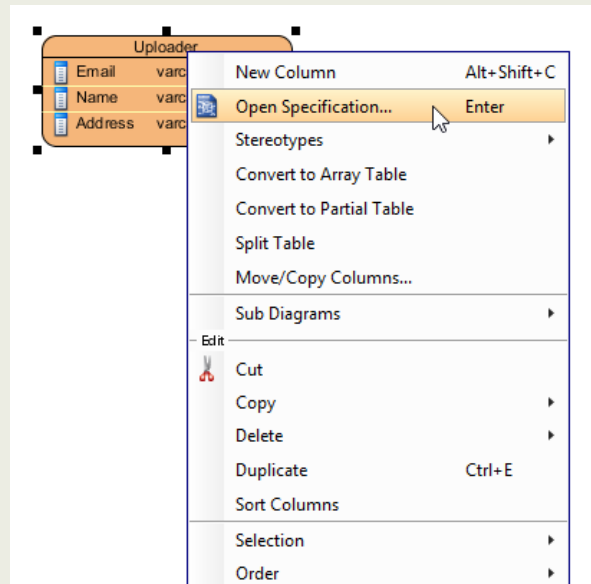
1. Right click on an entity.



Column in Detail

Defining index ■ Table space

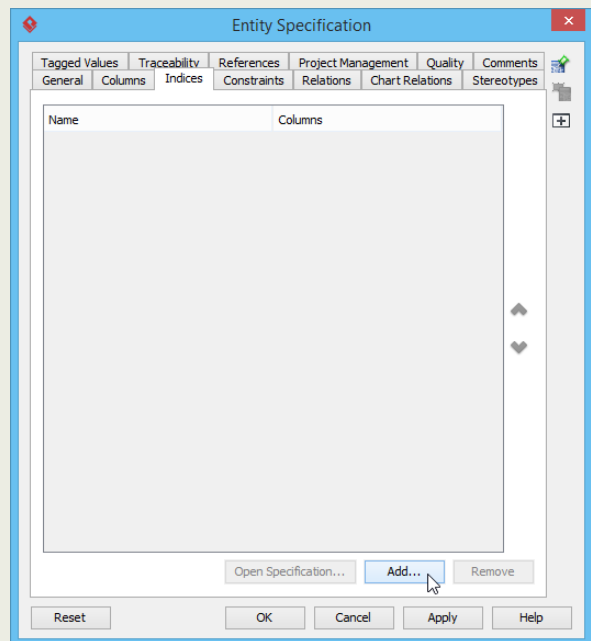
1. Right click on an entity.
2. Select **Open Specification...** from the popup menu.



Column in Detail

Defining index ■ Table space

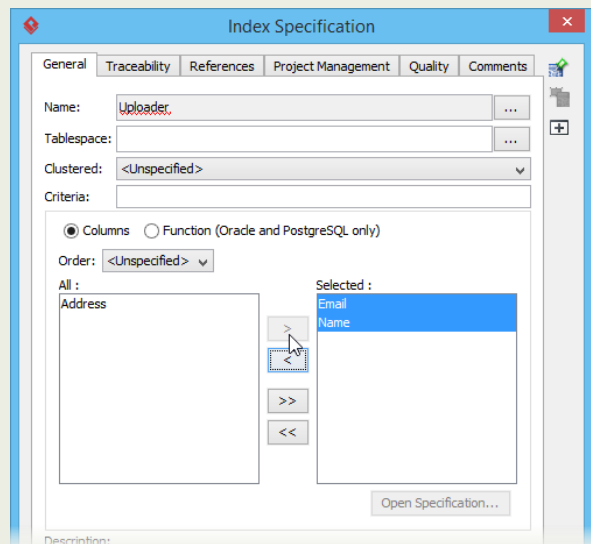
1. Right click on an entity.
2. Select **Open Specification...** from the popup menu.
3. Click **Add...** in the **Indices** tab.



Column in Detail

Defining index ■ Table space

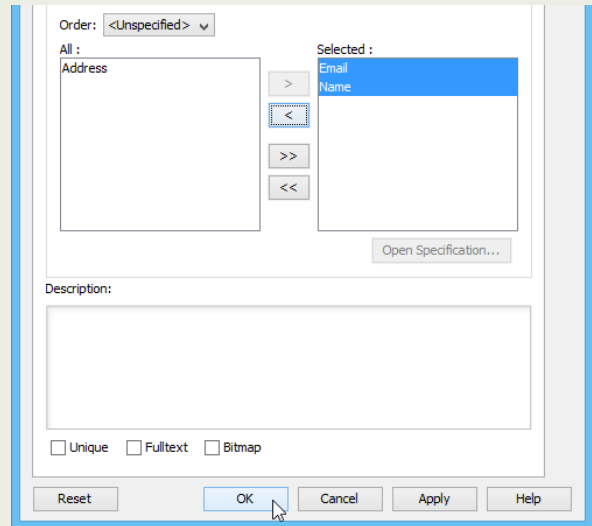
1. Right click on an entity.
2. Select **Open Specification...** from the popup menu.
3. Click **Add...** in the **Indices** tab.
4. Select the columns to be included in index and click **>**.



Column in Detail

Defining index ■ Table space

1. Right click on an entity.
2. Select **Open Specification...** from the popup menu.
3. Click **Add...** in the **Indices** tab.
4. Select the columns to be included in index and click **>**.
5. Click **OK** to confirm.



Column in Detail

Defining index ■ Column space

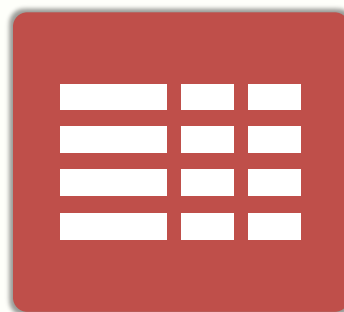
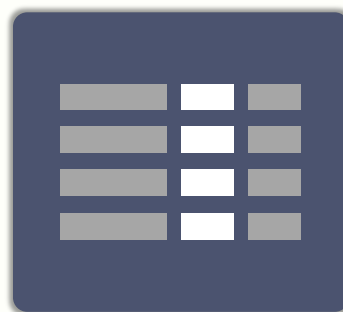


Table space
(Multiple columns in a table)

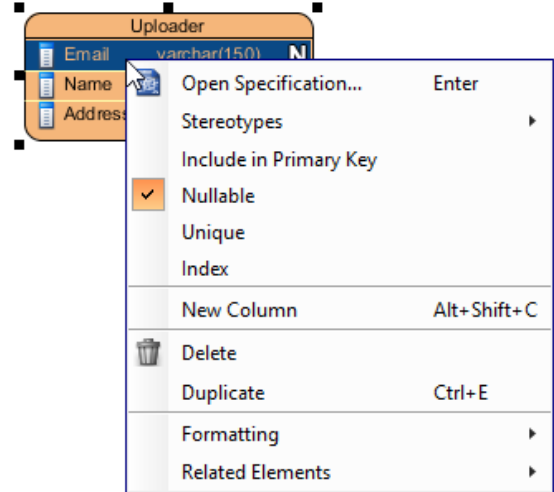


Column space
(Single column in a table)

Column in Detail

Defining index ■ Column space

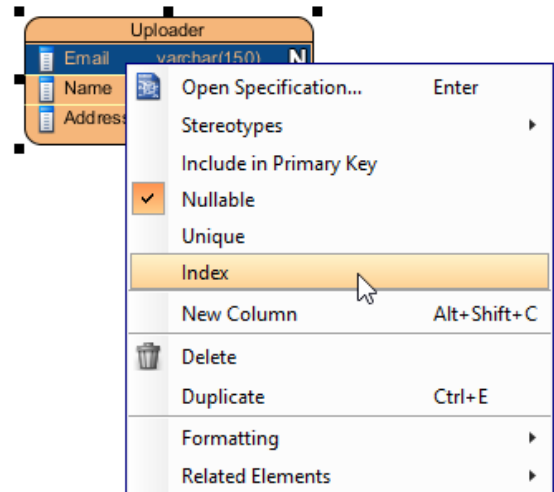
1. Right click on a column.



Column in Detail

Defining index ■ Column space

1. Right click on an entity.
2. Select **Index** from the popup menu.



Column in Detail

Summary

- How to open column specification
- Various properties of an column
- The use of index