[Intro music] Hi everyone. In this lecture, we will discuss about Oracle Database Management Tools. [No audio] In this lecture, you will learn how to perform the following, list the common Oracle database management tools, be familiar with using SQL*Plus, describe SQL Developer, describe EM Express, describe Oracle Enterprise Manager Cloud Control, and describe the Database Management Cloud Service. Let's get started. If you are already familiar with those tools, you can skip this lecture and go to the next one. Let's get started. [No audio] To manage Oracle databases, we have a list of management tools. There are pros and cons for each tool. So let's have a look at each one. SQL*Plus.

SQL*Plus is probably the most common tool used by DBAs to manage Oracle databases. It is a command line utility that allows the users to type the commands in the command prompt. It has been so popular because we can use it to run scripts and call it from script files. It is a lightweight utility that can be easily installed and used. On the other hand, it is tedious to display the data with so many columns in this utility. Formatting the retrieved data is a time consuming process. In addition to that, it is not easy to see the history of the executed commands, and quickly re-execute one of them. We can only use the history command which is again take some time. There are third party utilities by the way to fix this issue, but it is not supported by Oracle. SQL Developer. SQL Developer is a Java-based GUI interface that allows us to submit SQL statements to the database. Oracle has been improving the SQL Developer extensively in the recent years. It becomes now rich with advanced options and features. The retrieved data is displayed in a formatted tabular style. We can easily export the retreived data to external files, and use shortcuts to easily execute frequent operations. In the past, SQL Developer used to be shipped with Oracle database software. Currently, it is downloaded separately and is available for download in Oracle website. There is a command line utility that originally came with the SQL Developer, its name is SQLcl, but if you want to call it from the command line, you need to type SQL, only SQL, not SQLcl. This utility still comes with a SQL Developer, but it is still shipped with Oracle database software as well. The latest version of this utility can be downloaded separately. SQLcl is developed in first place to replace the SQL*Plus utility.

It covers the features that the SQL*Plus lacks, like inline editor, change management, command history, auto-complete object name or keywords using the tab key, with support in the same time to the SQL*Plus commands. It is a beautiful and feature utility. Oracle Enterprise Manager Database Express or EM Express. EM Express is a web-based GUI interface to monitor and partially manage a local database. We cannot use it to

connect to a remote database. This utility is nice for monitoring the database performance, but not anything else. Oracle Enterprise Manager Cloud Control or OEM. OEM is an advanced system that provides a web-based GUI interface to monitor and fully manage multiple databases. It does require a separate license and some learning curve is involved in using it. Database Management Cloud Service. Database Management Cloud Service is a service provided by Oracle Cloud to manage their on-premises and on the cloud databases.

You don't have to install anything. You just subscribe for the service, use it, pay for your usage, and that's it. [No audio] Third-party tools. The market provides some third-party tools to manage Oracle databases. They do make monitoring and managing Oracle databases much easier. But of course, you need to pay for their licenses. [No audio] To best of my knowledge, SQL*Plus is the most common tool used by the DBAs. It has been around since the very early Oracle versions, and still very commonly used among the DBAs. It is used to submit SQL statements and run scripts against Oracle databases. It has its own commands for formatting the displayed data. It could be used interactively or in scripts. As explained in the previous slide, SQLcl is a better tool. But for some reason, SQL*Plus is still more known and used among the DBAs than the SQLcl. We will stick with the SQL*Plus in the course practices.

Once you know about using SQL*Plus, you wouldn't face any problem switching to SQLcl if you want to. In this lecture, we will discuss about the various methods of invoking SQL*Plus and connecting to Oracle databases. In the next practice lecture, you will experience submitting common SQL*Plus commands. [No audio] To start SQL*Plus, use the command sqlplus. This command is saved in the bin directory under the Oracle Home directory. It is seen by the OS command line, because the bin directory is added to the path variable. The first code block in the slide demonstrates how to start SQL*Plus without connecting to any database, we use the switch /nolog. After invoking the SQL*Plus with this command, it opens the SQL*Plus command line prompt. We can then connect to any database using the connect command. The second code block in the slide demonstrates how to start the SQL*Plus with connecting to the local database as a normal user. We provide the user and its password separated by a forward slash. You can skip providing the password in the command line. In this case, the utility will ask you to enter the password after invoking it. While you type the password it doesn't appear in the display as you type it. If there are multiple database instances running in the system, the utility connects to the database pointed by the variable Oracle SID or Oracle SID. If the variable is not set, the connection fails. We can start the SQL*Plus with connecting to

the database as SYS user. SYS is the super user in Oracle databases that can do almost anything in the database. To connect to any Oracle Database as SYS, the connection must be as sysdba. as sysdba phrase must be accompanied with any connection attempt as SYS user. By the way, we are talking about SYS user, not system.
The third code block in the slide demonstrates how to invoke SQL*Plus with connecting to the local database as SYS. This command succeeds only if the OS users, who issues this command, belongs to the group dba. Normally only the software owner, which is typically Oracle, belongs to this group. Observe that in this connection attempt, we don't provide the SYS password. This method of connection is called OS Authentication and we will discuss about it in a separate lecture. If an OS user, who is not a member of the dba, wants to log in to the database as a SYS user, the user must know the SYS password and use it for connecting to the database. The connection format is demonstrated in the last code block in the slide. Again, the as sysdba phrase must be there. Those commands assumes that we run the SQL*Plus locally from the database server. But in fact, SQL*Plus can be invoked from the client machines as well, and it can connect to Oracle databases via the network connection.
We will cover this topic in a separate lecture. [No audio] SQL*Plus can be invoked from the command line to run a script and exec. The script could contain SQL statements, PL/SQL code execution, and SQL*Plus commands. By the way, SQL*Plus commands are not SQL statements. The first code block in the slide demonstrates an example of a script file that lists the departments in a specific location. To start SQL*Plus and run this script, we pass the script file name to the sqlplus command and add the @ symbol to prefix the file name. Observe that we don't have to type the file extension to its name when we pass it to the sqlplus command. This method of running scripts allows us to call SQL*Plus from shell scripts.
[No audio] Once we start SQL*Plus and successfully connect to the database, it will open its command line prompt and wait for the user to enter the SQL statement. The user just type directly the statement in the SQL*Plus command line. The statement can be entered as multiple lines, but it must end with the semicolon. After typing the semicolon and pressing Enter, the statement gets executed. [No audio] We can run the SQL scripts from the SQL*Plus command line. To achieve this task, we can either use the @ symbol followed by the script name that we want to execute, or we can submit the start command followed by the script name. The code block in the slide demonstrates two examples. Observe that typing the file extension is optional.

[No audio] SQL Developer is another fantastic tool to connect to Oracle databases and submit SQL statements to it. As explained earlier, it used to be shipped with Oracle database software. But starting from Oracle Database 19c, it doesn't come anymore with the software. We need to download it separately. Downloading and installing SQL Developer is super easy. Using it is even easier. We will download and installed SQL Developer in a practice lecture.
[No audio] As explained earlier, SQLcl is an a SQL*Plus alternative that comes with the SQL Developer. You can refer to the link provided in the slide to learn more about it. [No audio] Oracle EM Express is a web-based GUI interface for monitoring the local Oracle databases. It can be used to monitor only the local databases. We cannot use it to monitor remote databases. This tool is configured at the time of creating the database. Basically, it has three major pages. One for monitoring the performance, it's called Performance Hub. Another one for monitoring the tablespaces and the last one is called SQL Performance Analyzer. Despite it looks simple, do not underestimate its value.
EM Express provide something that is difficult to achieve using other tools, that is performance monitoring and troubleshooting. With EM Express, we can visualize the MTR database performance indicators. We can list the sessions that mostly make stress on the database. It also helps to narrow down to the statement that consumed much of the database resources compared to the other executed statements or which suffered from high weight events. Unfortunately, using the Performance Hub comes with a price. To be licensed for using it, you must own the Diagnostics Pack and the Tuning Pack licenses. As we learned earlier in the course, using those options are not part of the Oracle database Enterprise license.
[No audio] Oracle Enterprise Manager Cloud Control is a system management software that delivers centralized monitoring, administration, and lifecycle management functionality for the complete IT infrastructure, including systems running Oracle and non Oracle technologies. Of course, it includes managing Oracle databases. In simple words, Enterprise Manager Cloud Control offers a solution that allows you want to monitor and manage the complete Oracle IT infrastructure from a single console. As we saw in the Oracle database creation lecture, to be able to include an Oracle database system in an Enterprise Manager system, we must register the database in the system, and for that, we need to install an agent in the system. Needless to say, to use OEM, we need to buy its license. [No audio] Oracle provides a cloud service called Database Management Cloud Service. With this service, the dbas get a unified console for managing on-premises and the cloud databases. This may look similar to OEM, but this service is a specialized in

managing, monitoring, and troubleshooting only Oracle databases. This service provides advanced database fleet diagnostics, and tuning to troubleshoot issues and optimize the performance in Oracle databases. I highly recommend considering the service if you are managing high number of Oracle databases.
[No audio] And that's it for this lecture. By the end of this lecture, you should be able to perform the following, list common Oracle database management tools, be familiar with using SQL*Plus, describe SQL Developer, describe EM Express, describe Oracle Enterprise Manager Cloud Control, and describe Database Management Cloud Service. In the next lecture, you will get yourself familiar with using SQL*Plus. See you guys over there. Thank you very much. [No audio]
[Intro music] Hi everyone. In this lecture, we will discuss about Oracle Database Management Tools. [No audio] In this lecture, you will learn how to perform the following, list the common Oracle database management tools, be familiar with using SQL*Plus, describe SQL Developer, describe EM Express, describe Oracle Enterprise Manager Cloud Control, and describe the Database Management Cloud Service. Let's get started. If you are already familiar with those tools, you can skip this lecture and go to the next one. Let's get started. [No audio] To manage Oracle databases, we have a list of management tools. There are pros and cons for each tool. So let's have a look at each one. SQL*Plus. SQL*Plus is probably the most common tool used by DBAs to manage Oracle databases. It is a command line utility that allows the users to type the commands in the command prompt. It has been so popular because we can use it to run scripts and call it from script files. It is a lightweight utility that can be easily installed and used. On the other hand, it is tedious to display the data with so many columns in this utility. Formatting the retrieved data is a time consuming process. In addition to that, it is not easy to see the history of the executed commands, and quickly re-execute one of them. We can only use the history command which is again take some time. There are third party utilities by the way to fix this issue, but it is not supported by Oracle. SQL Developer. SQL Developer is a Java-based GUI interface that allows us to submit SQL statements to the database. Oracle has been improving the SQL Developer extensively in the recent years. It becomes now rich with advanced options and features. The retrieved data is displayed in a formatted tabular style. We can easily export the retreived data to external files, and use shortcuts to easily execute frequent operations. In the past, SQL Developer used to be shipped with Oracle database software. Currently, it is downloaded separately and is available for download in Oracle website. There is a command line utility that originally came with the SQL Developer, its name is SQLcl, but if you want to call it from the command line, you need to type SQL, only SQL, not SQLcl. This utility still comes with a SQL Developer, but it is still shipped with Oracle database software as well. The latest version of this utility can be downloaded separately. SQLcl is developed in first place to replace the SQL*Plus utility. It covers the features that the SQL*Plus lacks, like inline editor, change management, command history, auto-complete object name or keywords using the tab key, with support in the same time to the SQL*Plus commands. It is a beautiful and feature utility.
Oracle Enterprise Manager Database Express or EM Express. EM Express is a web-based GUI interface to monitor and partially manage a local database. We cannot use it to connect to a remote database. This utility is nice for monitoring the database performance, but not anything else. Oracle Enterprise Manager Cloud Control or OEM. OEM is an advanced system that provides a web-based GUI interface to monitor and fully manage

multiple databases. It does require a separate license and some learning curve is involved in using it. Database Management Cloud Service. Database Management Cloud Service is a service provided by Oracle Cloud to manage their on-premises and on the cloud databases. You don't have to install anything. You just subscribe for the service, use it, pay for your usage, and that's it. [No audio] Third-party tools. The market provides some third-party tools to manage Oracle databases. They do make monitoring and managing Oracle databases much easier. But of course, you need to pay for their licenses. [No audio] To best of my knowledge, SQL*Plus is the most common tool used by the DBAs. It has been around since the very early Oracle versions, and still very commonly used among the DBAs. It is used to submit SQL statements and run scripts against Oracle databases. It has its own commands for formatting the displayed data. It could be used interactively or in scripts. As explained in the previous slide, SQLcl is a better tool. But for some reason, SQL*Plus is still more known and used among the DBAs than the SQLcl. We will stick with the SQL*Plus in the course practices. Once you know about using SQL*Plus, you wouldn't face any problem switching to SQLcl if you want to. In this lecture, we will discuss about the various methods of invoking SQL*Plus and connecting to Oracle databases. In the next practice lecture, you will experience submitting common SQL*Plus commands. [No audio] To start SQL*Plus, use the command sqlplus. This command is saved in the bin directory under the Oracle Home directory. It is seen by the OS command line, because the bin directory is added to the path variable. The first code block in the slide demonstrates how to start SQL*Plus without connecting to any database, we use the switch /nolog. After invoking the SQL*Plus with this command, it opens the SQL*Plus command line prompt. We can then connect to any database using the connect command. The second code block in the slide demonstrates how to start the SQL*Plus with connecting to the local database as a normal user. We provide the user and its password separated by a forward slash. You can skip providing the password in the command line. In this case, the utility will ask you to enter the password after invoking it. While you type the password it doesn't appear in the display as you type it. If there are multiple database instances running in the system, the utility connects to the database pointed by the variable Oracle SID or Oracle SID. If the variable is not set, the connection fails. We can start the SQL*Plus with connecting to the database as SYS user. SYS is the super user in Oracle databases that can do almost anything in the database. To connect to any Oracle Database as SYS, the connection must be as sysdba. as sysdba phrase must be accompanied with any connection attempt as SYS user. By the way, we are talking about SYS user, not system. The third code block in the slide demonstrates how to invoke SQL*Plus with connecting to the local database as SYS. This command succeeds only if the OS users, who issues this command, belongs to the group dba. Normally only the software owner, which is typically Oracle, belongs to this group. Observe that in this connection attempt, we don't provide the SYS password. This method of connection is called OS Authentication and we will discuss about it in a separate lecture. If an OS user, who is not a member of the dba, wants to log in to the database as a SYS user, the user must know the SYS password and use it for connecting to the database. The connection format is demonstrated in the last code block in the slide. Again, the as sysdba phrase must be there. Those commands assumes that we run the SQL*Plus locally from the database server. But in fact, SQL*Plus can be invoked from the client machines as well, and it can connect to Oracle databases via the network connection. We will cover this topic in a separate lecture. [No audio] SQL*Plus can be invoked from the command line to run a script and exec. The script could contain SQL statements, PL/SQL code execution, and SQL*Plus commands. By the way, SQL*Plus commands are not SQL statements. The first code block in the slide demonstrates an example of a script file that lists the departments in a specific location. To start SQL*Plus and run this script, we pass the script file name to the

sqlplus command and add the @ symbol to prefix the file name. Observe that we don't have to type the file extension to its name when we pass it to the sqlplus command. This method of running scripts allows us to call SQL*Plus from shell scripts. [No audio] Once we start SQL*Plus and successfully connect to the database, it will open its command line prompt and wait for the user to enter the SQL statement. The user just type directly the statement in the SQL*Plus command line. The statement can be entered as multiple lines, but it must end with the semicolon. After typing the semicolon and pressing Enter, the statement gets executed. [No audio] We can run the SQL scripts from the SQL*Plus command line. To achieve this task, we can either use the @ symbol followed by the script name that we want to execute, or we can submit the start command followed by the script name. The code block in the slide demonstrates two examples. Observe that typing the file extension is optional. [No audio] SQL Developer is another fantastic tool to connect to Oracle databases and submit SQL statements to it. As explained earlier, it used to be shipped with Oracle database software. But starting from Oracle Database 19c, it doesn't come anymore with the software. We need to download it separately. Downloading and installing SQL Developer is super easy. Using it is even easier. We will download and installed SQL Developer in a practice lecture. [No audio] As explained earlier, SQLcl is an a SQL*Plus alternative that comes with the SQL Developer. You can refer to the link provided in the slide to learn more about it. [No audio] Oracle EM Express is a web-based GUI interface for monitoring the local Oracle databases. It can be used to monitor only the local databases. We cannot use it to monitor remote databases. This tool is configured at the time of creating the database. Basically, it has three major pages. One for monitoring the performance, it's called Performance Hub. Another one for monitoring the tablespaces and the last one is called SQL Performance Analyzer. Despite it looks simple, do not underestimate its value. EM Express provide something that is difficult to achieve using other tools, that is performance monitoring and troubleshooting. With EM Express, we can visualize the MTR database performance indicators. We can list the sessions that mostly make stress on the database. It also helps to narrow down to the statement that consumed much of the database resources compared to the other executed statements or which suffered from high weight events. Unfortunately, using the Performance Hub comes with a price. To be licensed for using it, you must own the Diagnostics Pack and the Tuning Pack licenses. As we learned earlier in the course, using those options are not part of the Oracle database Enterprise license. [No audio] Oracle Enterprise Manager Cloud Control is a system management software that delivers centralized monitoring, administration, and lifecycle management functionality for the complete IT infrastructure, including systems running Oracle and non Oracle technologies. Of course, it includes managing Oracle databases. In simple words, Enterprise Manager Cloud Control offers a solution that allows you want to monitor and manage the complete Oracle IT infrastructure from a single console. As we saw in the Oracle database creation lecture, to be able to include an Oracle database system in an Enterprise Manager system, we must register the database in the system, and for that, we need to install an agent in the system. Needless to say, to use OEM, we need to buy its license. [No audio] Oracle provides a cloud service called Database Management Cloud Service. With this service, the dbas get a unified console for managing on-premises and the cloud databases. This may look similar to OEM, but this service is a specialized in managing, monitoring, and troubleshooting only Oracle databases. This service provides advanced database fleet diagnostics, and tuning to troubleshoot issues and optimize the performance in Oracle databases. I highly recommend considering the service if you are managing high number of Oracle databases. [No audio] And that's it for this lecture. By the end of this lecture, you should be able to perform the following, list common Oracle database management tools, be familiar with using SQL*Plus, describe SQL

Developer, describe EM Express, describe Oracle Enterprise Manager Cloud Control, and describe Database Management Cloud Service. In the next lecture, you will get yourself familiar with using SQL*Plus. See you guys over there. Thank you very much. [No audio] [Intro music] Hi everyone. In this lecture, we will discuss about Oracle Database Management Tools. [No audio] In this lecture, you will learn how to perform the following, list the common Oracle database management tools, be familiar with using SQL*Plus, describe SQL Developer, describe EM Express, describe Oracle Enterprise Manager Cloud Control, and describe the Database Management Cloud Service. Let's get started. If you are already familiar with those tools, you can skip this lecture and go to the next one. Let's get started. [No audio] To manage Oracle databases, we have a list of management tools. There are pros and cons for each tool. So let's have a look at each one. SQL*Plus. SQL*Plus is probably the most common tool used by DBAs to manage Oracle databases. It is a command line utility that allows the users to type the commands in the command prompt. It has been so popular because we can use it to run scripts and call it from script files. It is a lightweight utility that can be easily installed and used. On the other hand, it is tedious to display the data with so many columns in this utility. Formatting the retrieved data is a time consuming process. In addition to that, it is not easy to see the history of the executed commands, and quickly re-execute one of them. We can only use the history command which is again take some time. There are third party utilities by the way to fix this issue, but it is not supported by Oracle. SQL Developer. SQL Developer is a Java-based GUI interface that allows us to submit SQL statements to the database. Oracle has been improving the SQL Developer extensively in the recent years. It becomes now rich with advanced options and features. The retrieved data is displayed in a formatted tabular style. We can easily export the retreived data to external files, and use shortcuts to easily execute frequent operations. In the past, SQL Developer used to be shipped with Oracle database software. Currently, it is downloaded separately and is available for download in Oracle website. There is a command line utility that originally came with the SQL Developer, its name is SQLcl, but if you want to call it from the command line, you need to type SQL, only SQL, not SQLcl. This utility still comes with a SQL Developer, but it is still shipped with Oracle database software as well. The latest version of this utility can be downloaded separately. SQLcl is developed in first place to replace the SQL*Plus utility. It covers the features that the SQL*Plus lacks, like inline editor, change management, command history, auto-complete object name or keywords using the tab key, with support in the same time to the SQL*Plus commands. It is a beautiful and feature utility. Oracle Enterprise Manager Database Express or EM Express. EM Express is a web-based GUI interface to monitor and partially manage a local database. We cannot use it to connect to a remote database. This utility is nice for monitoring the database performance, but not anything else. Oracle Enterprise Manager Cloud Control or OEM. OEM is an advanced system that provides a web-based GUI interface to monitor and fully manage multiple databases. It does require a separate license and some learning curve is involved in using it. Database Management Cloud Service. Database Management Cloud Service is a service provided by Oracle Cloud to manage their on-premises and on the cloud databases. You don't have to install anything. You just subscribe for the service, use it, pay for your usage, and that's it. [No audio] Third-party tools. The market provides some third-party tools to manage Oracle databases. They do make monitoring and managing Oracle databases much easier. But of course, you need to pay for their licenses. [No audio] To best of my knowledge, SQL*Plus is the most common tool used by the DBAs. It has been around since the very early Oracle versions, and still very commonly used among the DBAs. It is used to submit SQL statements and run scripts against Oracle databases. It has its own commands for formatting the displayed data. It could be used interactively or in scripts.

As explained in the previous slide, SQLcl is a better tool. But for some reason, SQL*Plus is still more known and used among the DBAs than the SQLcl. We will stick with the SQL*Plus in the course practices. Once you know about using SQL*Plus, you wouldn't face any problem switching to SQLcl if you want to. In this lecture, we will discuss about the various methods of invoking SQL*Plus and connecting to Oracle databases. In the next practice lecture, you will experience submitting common SQL*Plus commands. [No audio] To start SQL*Plus, use the command sqlplus. This command is saved in the bin directory under the Oracle Home directory. It is seen by the OS command line, because the bin directory is added to the path variable. The first code block in the slide demonstrates how to start SQL*Plus without connecting to any database, we use the switch /nolog. After invoking the SQL*Plus with this command, it opens the SQL*Plus command line prompt. We can then connect to any database using the connect command. The second code block in the slide demonstrates how to start the SQL*Plus with connecting to the local database as a normal user. We provide the user and its password separated by a forward slash. You can skip providing the password in the command line. In this case, the utility will ask you to enter the password after invoking it. While you type the password it doesn't appear in the display as you type it. If there are multiple database instances running in the system, the utility connects to the database pointed by the variable Oracle SID or Oracle SID. If the variable is not set, the connection fails. We can start the SQL*Plus with connecting to the database as SYS user. SYS is the super user in Oracle databases that can do almost anything in the database. To connect to any Oracle Database as SYS, the connection must be as sysdba. as sysdba phrase must be accompanied with any connection attempt as SYS user. By the way, we are talking about SYS user, not system. The third code block in the slide demonstrates how to invoke SQL*Plus with connecting to the local database as SYS. This command succeeds only if the OS users, who issues this command, belongs to the group dba. Normally only the software owner, which is typically Oracle, belongs to this group. Observe that in this connection attempt, we don't provide the SYS password. This method of connection is called OS Authentication and we will discuss about it in a separate lecture. If an OS user, who is not a member of the dba, wants to log in to the database as a SYS user, the user must know the SYS password and use it for connecting to the database. The connection format is demonstrated in the last code block in the slide. Again, the as sysdba phrase must be there. Those commands assumes that we run the SQL*Plus locally from the database server. But in fact, SQL*Plus can be invoked from the client machines as well, and it can connect to Oracle databases via the network connection. We will cover this topic in a separate lecture. [No audio] SQL*Plus can be invoked from the command line to run a script and exec. The script could contain SQL statements, PL/SQL code execution, and SQL*Plus commands. By the way, SQL*Plus commands are not SQL statements. The first code block in the slide demonstrates an example of a script file that lists the departments in a specific location. To start SQL*Plus and run this script, we pass the script file name to the sqlplus command and add the @ symbol to prefix the file name. Observe that we don't have to type the file extension to its name when we pass it to the sqlplus command. This method of running scripts allows us to call SQL*Plus from shell scripts. [No audio] Once we start SQL*Plus and successfully connect to the database, it will open its command line prompt and wait for the user to enter the SQL statement. The user just type directly the statement in the SQL*Plus command line. The statement can be entered as multiple lines, but it must end with the semicolon. After typing the semicolon and pressing Enter, the statement gets executed. [No audio] We can run the SQL scripts from the SQL*Plus command line. To achieve this task, we can either use the @ symbol followed by the script name that we want to execute, or we can submit the start command followed by the script name. The code block in the slide demonstrates two examples. Observe that typing the file extension is

optional. [No audio] SQL Developer is another fantastic tool to connect to Oracle databases and submit SQL statements to it. As explained earlier, it used to be shipped with Oracle database software. But starting from Oracle Database 19c, it doesn't come anymore with the software. We need to download it separately. Downloading and installing SQL Developer is super easy. Using it is even easier. We will download and installed SQL Developer in a practice lecture. [No audio] As explained earlier, SQLcl is an a SQL*Plus alternative that comes with the SQL Developer. You can refer to the link provided in the slide to learn more about it. [No audio] Oracle EM Express is a web-based GUI interface for monitoring the local Oracle databases. It can be used to monitor only the local databases. We cannot use it to monitor remote databases. This tool is configured at the time of creating the database. Basically, it has three major pages. One for monitoring the performance, it's called Performance Hub. Another one for monitoring the tablespaces and the last one is called SQL Performance Analyzer. Despite it looks simple, do not underestimate its value. EM Express provide something that is difficult to achieve using other tools, that is performance monitoring and troubleshooting. With EM Express, we can visualize the MTR database performance indicators. We can list the sessions that mostly make stress on the database. It also helps to narrow down to the statement that consumed much of the database resources compared to the other executed statements or which suffered from high weight events. Unfortunately, using the Performance Hub comes with a price. To be licensed for using it, you must own the Diagnostics Pack and the Tuning Pack licenses. As we learned earlier in the course, using those options are not part of the Oracle database Enterprise license. [No audio] Oracle Enterprise Manager Cloud Control is a system management software that delivers centralized monitoring, administration, and lifecycle management functionality for the complete IT infrastructure, including systems running Oracle and non Oracle technologies. Of course, it includes managing Oracle databases. In simple words, Enterprise Manager Cloud Control offers a solution that allows you want to monitor and manage the complete Oracle IT infrastructure from a single console. As we saw in the Oracle database creation lecture, to be able to include an Oracle database system in an Enterprise Manager system, we must register the database in the system, and for that, we need to install an agent in the system. Needless to say, to use OEM, we need to buy its license. [No audio] Oracle provides a cloud service called Database Management Cloud Service. With this service, the dbas get a unified console for managing on-premises and the cloud databases. This may look similar to OEM, but this service is a specialized in managing, monitoring, and troubleshooting only Oracle databases. This service provides advanced database fleet diagnostics, and tuning to troubleshoot issues and optimize the performance in Oracle databases. I highly recommend considering the service if you are managing high number of Oracle databases. [No audio] And that's it for this lecture. By the end of this lecture, you should be able to perform the following, list common Oracle database management tools, be familiar with using SQL*Plus, describe SQL Developer, describe EM Express, describe Oracle Enterprise Manager Cloud Control, and describe Database Management Cloud Service. In the next lecture, you will get yourself familiar with using SQL*Plus. See you guys over there. Thank you very much. [No audio] [Intro music] Hi everyone. In this lecture, we will discuss about Oracle Database Management Tools. : Added to Selection. Press [CTRL + S] to save as a note : Added to Selection. Press [CTRL + S] to save as a note : Added to Selection. Press [CTRL + S] to save as a note