Hi, I’m happy to be here

# About Me:

I’m Pranay Yara. I'm from India, I moved to Australia in 2010 when I was 8 years old and since then I've been living in Adelaide.

**So, a bit more about me is:**

* I am a **recent graduate** from the University of Adelaide.
* During my time there, I completed a Bachelor’s of Computer Science, developed a strong foundation in **programming and software development**.
* Major: **Data Science** - **data analysis and statistical modelling**.
* After my bachelors, **curious about cyber security** due to a lot of **cyber crimes** in the news.
* So, Joined Adelaide University’s Cyber Security bootcamp, opportunity to **learn from two mentors for 6 months.**
* And After graduating my degree and bootcamp, I came across this wonderful opportunity **to work for DXC** and **learn mainframes**.
* So far, **training and support from genspark** has been outstanding.
* Lastly, two attributes about me are; I am a **continuous learner** and a **problem solver**.

# About Training:

* The training **has been great,** of course there were **challenges** at the start such as the **style of teaching** which worked for previous mainframe cohorts, wasn’t working for us but collaboratively with everyones opinions we were able to **move forward with a customised training style.**
* Besides that, I learnt about **JCL, COBOL, CICS and currently, DB2.** Also, I’m really happy about the training because of my team - they are **nice, helpful and** I would love to work with them in the future.
* We are **very diverse** and we have **a collaborative environment** that keeps us on the **same page.**

# About DXC?

* DXC is an **IT services company,** uses the **power of technology** to build better futures for **its customers, colleagues and communities.**
* DXC is **modernising IT, optimising data architectures, and ensuring security and scalability across public, private and hybrid clouds.**
* **The world's largest companies** and **public sector organisations** trust DXC to **deploy** services to drive new levels of performance, competitiveness, and customer experience.
* Mission: **Deliver mission critical IT services for clients.**
* Values: **Deliver** outcomes, **Doing** the right thing, **Care** for each other, **Collaborate** and **support communities.**
* WIth this mainframes program, I understand that many mainframe **skills are niche in the current marketplace and current mainframe** staff have 20-30 years of experience that needs to be passed to newer generations of mainframers. This program aligns with DXC’s values.

# Why DXC?

* What I really like about DXC **and why I** want to be part of DXC is due to **the power of its people** as their presence is in over 60 countries, 130k employees, and 240 customers in just the Fortune500.
* **DXC’s learning platform** for its employees and community programs such as the **my mainframes program** and **the dandelion program** for neurodivergent individuals is admiring.
* Even the DXC culture I saw from **Dalibor, Micheal and Jacob** made DXC seem like a perfect workplace for someone like me - who wants to **Trust, Transform & Thrive.**
* Actually a story, I have known DXC and the office in Adelaide **since 2018. I** was in my high school looking for degrees and I found a unisa degree in collaboration with a pathway to DXC. That time my father advised me to pursue the degree to join DXC because that's how highly he saw DXC. **And the office** is only 5 mins away from my home, but unfortunately just as I was about to enter university that degree was discontinued due to too much demand. Now, Here I am, 6 years later with another chance to join DXC.

# About DXC Mainframes?

I noticed in the meet and greet **mainframe modernisation** was a key topic. So, I researched and read an **insightful article on the DXC website by Murali,** explaining about the mainframe modernisation. Some of the things include:

* Companies with mainframes are **facing the difficult task of figuring out the best ways** to modernise mainframes in order to stay competitive.
* The **two primary reasons,** DXC customers want to modernise their mainframes is to **first reducing the total cost of ownership,** especially due to the rising costs of infrastructure and software. The **second reason is to gain flexibility** as launching new products can be difficult and time-consuming with legacy systems.
* However, the challenge is that **every mainframe is unique,** so it's crucial to identify the right modernisation strategy to achieve desired business outcomes.
* I also read about some of the **key factors for successful** mainframe modernisation:

1. First one, being prepared for **adopting and adapting to change.** For example, being ready to move onto cloud-native technologies post-modernisation.
2. Ensuring **stakeholder alignment** for technical goals and business outcomes.
3. **Identifying the right landing zone and transformation path.** Whether on-premises or public cloud or hybrid cloud.
4. Having the right **mainframe staff.**
5. **Talking** to those who have **completed mainframe modernisation** and asking them questions. (What does success mean to you? Pain points they went through to achieve success and lessons learnt.)

I also read about **DXC’s approach** which is to perform a 360-degree review of a client's future mainframe options, then identify a solution that protects existing investments while creating a path to modernising and transforming the client's mainframe estate.

I read an article about mainframe modernisation on the DXC website:  
I found out about:  
Why clients want to modernise   
What are the success factors for mainframe modernisation

And DXC’s approach to mainframe modernisation

# Time Management / Work Load:

**I use priority lists** which is a **technical concept from computer science,** where I allocate dedicated time slots and give priority for each task based on their important dates.

* Yes, I can tell you about a time.
* **Situation**: final semester, challenge: **managing an overloaded coursework, while maintaining strong grades**.
* **Task**: Manage six subjects rather than the regular four because I had some prerequisite subjects that I didn’t do during covid and online classes. So it **was extremely difficult to be organised or even get work completed on time**.
* **Action**: To address this, **priority list**, **allocating dedicated time slots for each subject based on their important dates** such as exams, assignment deadlines and presentations.
* **Result**: and By being **dynamic**, I **stayed organised and managed my workload efficiently**, **Averaging a distinction** for four out of the six subjects.
* Also, during this time, I was **resilient and committed to seeing ideas come to success** by not giving up even though times were difficult - which was another factor that helped me to **pass all the subjects**.
* So that is the time and I always learn from my lessons so even now, all the priority, and organising skills I developed from that time I still use **in my current tasks**.

# Team Work:

**When working** in a team, I **try to show** initiative, proactiveness and collaboration. I make sure there is **open communication, mutual support and friendship.**

* I can definitely tell you about a time.
* **Situation:** During a defensive security project.
* **Task:** My team had to **analyse logs and implement automated alerts to stop cyber attacks**. Everything was going well, however, One of my team members, **Alex preferred working alone and would only talk if you talk to him specifically** in our group discussions. We also had a presentation which we **needed Alex to be part of**.
* **Action:** To solve this, I initiated check ups with Alex to **discuss project goals, share insights, and just see how he's doing - trying to be his friend**.
* **With this Open communication**, I was able to find out the reason Alex **lacked collaboration. It was due to his language barrier**. Knowing this I **took more care** into making sure Alex understood our team discussions. This included **speaking more** **slowly, clearly, and using shared documents**.
* **Result**: All of this led to **Alex being more comfortable with his language barrier**. He **started to give his inputs** in discussions. He also **did his part** for the presentation and **became a good friend** of mine. Ultimately our **project was approved** by our mentor | who is Byte’s Information Security Lead.
* So, yeah that’s a time where I showcased not only team working skills but also **initiative, proactiveness** and mutual support for my team's success.

# Adaptability / Results Oriented:

* **Yes, I can tell you about a time.**
* **Situation**: For my penetration testing project, I needed to use cyber tools andone of the tools was called **Nmap. - It's a network scanner, which I found confusing to use** because you need other tools to further test the information from nmap scans.
* Also, I **only had three days** of access for the project.
* **Task**: So, I had to master nmap as fast as possible to contribute to project success and improve my penetration testing skills.
* **Action**: As I didn't have much time, rather than reviewing lengthy videos or reading textbooks, I tried to be innovative in my learning **and I chose** to use an **interactive learning method of Capture The Flag** challenges on **Hack The Box** to master Nmap.
* There were **real-world scenarios** used in these challenges which allowed me to gain practical knowledge, while making the learning process both engaging and efficient.
* **Result**: As for the result, I found this method to be very **innovative because I significantly cut down the learning time.** Also, I used nmap at a high level in my project, my group identified all the vulnerabilities findable with nmap scan.

# Problem Solving / Mistake:

To summarise, how I solve problems is by breaking down our progress into parts and trying to identify the root cause. I also use critical thinking to pick up on any key observations and then make a concluding decision to address the problem.

* **A time is**
* **Situation & Task**: During a cyber security defensive security project using **Splunk**. My group cruised through all the tasks.
* However, when we were writing the project brief. I noticed my data values were different to my group’s despite having the same graphs.
* **Action**: **to solve this issue,** I first tried to **identify the root cause** and just **break down my progress into parts** to get an idea where the problem occurred.
* Although I couldn’t find the exact cause. **Using my critical-thinking**, I had some observations.
* I noticed that I might have **uploaded the data multiple times**, so the values got duplicated, leading to me having different total values but similar maps to my group. This unfortunately is a bug which is common in virtual machines.
* So, we went ahead with the most common results for the project brief. We also checked our findings with another group for accuracy.
* **Result**: In the end, using the common results based on our observations, turned out to be a good thing. As our project received **full marks** and was **approved by Byte's Information Security Lead**.

# **Diversity, Equity and Inclusion:** Yes, these are very important to me.

* Firstly I believe **diverse teams** have **a range of different perspectives** which ultimately brings more success.
* Secondly, equity, **everyone is equal** - everyone in the team should have the same access to resources and help.
* Thirdly, inclusive culture is really important to me because this allows people to work at their best and happily - when everyone feels included.

# Challenge at DXC.

Two challenges I can think off are:

As mainframes frames are unique and DXC has a wide range of client base - learning about the best strategies to support dxc delivery service to clients seems a bit challenging.

Mainframes has been here for a long time and it's still constantly evolving, and keeping up with new updates, software versions, and also catching up again seems a bit challenging.

But I am ready to take on the challenge.

# 

# **Project Management:**

* My most recent project was about **Developing and designing a custom monitoring solution using Splunk Enterprise.**
* Which involved working in a group for a week, to set thresholds and baselines **to generate automated alerts** into the manager’s email - **On any suspicious activity detected** in the servers.
* **My role as a SOC Analyst,** was to analyse previous attack logs, design SPL queries to create dashboards with single-value visualisations, multiple-value visualisations and geographical maps to provide in-depth analysis of attack signatures, for example; the sort of activity during the attack, the time frames of suspicious activity and the location where the attacks were from.
* I managed these tasks with my team - **it was a collaborative environment - everyone had their set tasks and we all worked in the same call along with shared documents. \**^<-* how I work in a team*\****
* In the end, we submitted the project on time, we also found out what the attack was **(it was a brute force - from ukraine)** and we presented a monitoring solution that was approved by our mentor who is an Information Security Manager.

# **Customer Service:**

* I don't have any customer serving experience but how I would approach it is: always being calm, trying to understand their perspective and providing them with solutions swiftly - while keeping them updated at all stages.

# Weaknesses:

* One of my weaknesses has been public speaking. Due to **my lack of experience** in this area, I **often felt nervous and unsure** when presenting. Recognising the importance of communication skills, especially for a job, I wanted to get more experience in presenting.
* Around the same time, there was a subject in university called professional practice, which had **2 hour in-class discussions every week and a final project with a 10 minute presentation.** I saw this as an opportunity to improve my public speaking and I enrolled into the subject. In it I made sure to attend all the classes and participate in presenting. This resulted in me getting a **distinction** which **boosted my confidence**, and I’m **quite comfortable** with public speaking now.
* And I use this same concept for any weaknesses I sense, I practise and get experience till it becomes a strength of mine.

Another weakness - time consuming due to focus on quality

# **Strength:**

* My diverse skillset is my strength. As I have a passion for continuous learning, I have learnt a lot of different domains during my time at university such as programming, software development, data science & analysis to cyber security. All these technologies complement each other. For example, this week when we were learning db2, my prior knowledge of creating sql databases has helped me understand db2 concepts easier.

# Decision Making / Change:

change from data science to cyber and IT

* A time would be my change from data science to cyber security.
* In my final year when I was majoring in data science, I had a cyber security subject as my computer science elective.
* During this, I liked the work I was doing in cyber security rather than the work in data science. Cyber security was fresh, every week we would learn a new tool, there was capture the flags, and it was different to programming.
* Which was the main problem in data science, I had to learn python, and also a lot of maths concepts from textbooks which was similar to my first two years. I had to learn C++ and read textbooks to understand programming concepts.
* Another reason, was one of the main reasons I chose the entire degree was for a subject related to big data. However, when I got to the stage of selecting that subject it was moved to only masters degree students.
* So yeah during that time cyber was something new for me and I instantly became passionate about it and also there were also cyber attacks in the news, which was another motivating factor driving me to enrolling into Adelaide universities cyber security bootcamp.

# Negative Feedback?

* I’m open to negative feedback. I look at it as a way to improve myself. I will just try to understand their perspective and be more adaptable.

# Tackle problems you can't handle?

* I would first try to assess the problem - see what the challenge is and try to find solutions. And if that doesn’t work I will try to consult my team members and mentors to get their advice on it.

Q. What Other teams will I be interacting **with apart from the mainframe teams? Maybe the business teams or software teams.**

# 

# **z/OS**

## What is mainframe?

a high-performance computer used for large-scale, compute-intensive purposes and tasks - find out what they mean for dxc.

* **z/OS** is a zero down time operating system. So, systems using z/OS have minimal downtime, aiming for continuous availability.
* In z/OS everything must be communicated as a job. z/OS consists of many sub systems:
* There are two types of language in mainframes - **application language** (COBOL): building logic / commercial - business needs, insurance, retail market, airlines - any applications that are daily / customer needs. Opposite to that we have **control language** (JCL): is not a commercial, it's built to interrupt / will help communicate with the operating system not to build your commercial applications.
* Scheduling + JES subsystem
* Write memory about memory

## TSO

* **Ispf** is an application in **tso** (tso is a bigger thing, within tso is ispf)
* **Datasets** have two purposes: either to **store raw data** (collection of records - a **record** is (ID)(NAME)(INFO) ‘01BOBENGINEER80000’) **or to write code.**
* Two types of datasets: **non VSAM** and VSAM (virtual storage access method).
* In **non VSAM** dataset, there are two types of datasets: **PDS** (Partitioned Dataset) and **PS** (Physical Sequential).
* A **PDS** is a collection of members. Each member can have jcl code.
* A **PS** file is an independent or scratch dataset, which either acts as an input or output file to the program.
* Both PDS and PS files are **sequential**, what is a sequential file? where records are accessed 1 by 1.
* **Resource** is anything requested by the program (mostly input / output datasets)

# **JCL**

**JCL is a** control language that acts as an interface between application programs and the operating system.

**JCL Statements: JOB** (max 255)**, EXEC** (max 255)**, DD** (3273)

**Parameters for JOB statement:** Positional parameters don't have = while keyword have it.

**Positional Parameters:** ACCT (Account Information), NAME

**Keyword Parameters:**

* MSGLEVEL -specifies the messages generated in the output. Default is (1,1).
* MSGCLASS - specifies the job output class. Class can be from A-Z, 1-9.
* TYPRUN - = SCAN checks for JCL syntax & format errors. Or = HOLD for later exec.
* PRTY - assigns priority to the jobs. 1-15, 15 gets the highest priority.
* CLASS - an input queue for holding jobs awaiting execution. A to Z and 0 to 9.
* RESTART - specifies the step name from which the job needs to be restarted.
* REGION - maximum amount of memory that can be allotted for the job. Optimal= 0m.
* TIME - maximum time that the JOB can take to run.
* NOTIFY - notifies you when a job is submitted and tells you the max cc code.
* **COND** - is for decision making, its like an if statement **syntax:** is cond=(relation condition,relation operator,previous step /stepname) - Can be coded in JOB or EXEC
* **Conditions** are **boolean** bytes so true or false. **false=execute**, **true=bypass/skip**.
* **0** is a sucess, **4** is a warning, **8** is syntax error, **12** dataset error, **16** utilility control statement errors.
* **cond=only,** means that previous step must be abended then step will get executed. **cond=even,** run steps even though any of the previous steps has abended.

**DD Statement:**  
 DD DSN='FILE.PATH',DISP=(NEW,CALTG,DELETE)   
 UNIT=3390,VOL=SER=DEVHD1,SPACE=(TRK,(5,5)),   
 DCB (DATA CONTROL BLOCK) =(RECFM=FB,LRECL=80,BLKSIZE=800,DSORG=PS)

Special dd names: SYSUT1 SYSUT2, SYSIN SYSOUT

Sysout =\* prints output  
Sysprint =\* prints error messages   
Sysin = \*/dummy / dummy if here is no follow up or input data otherwise \*

## JCL Utilities

**IEFBR14: DUMMY UTILITY**

* used for creating a ps/pds dataset - disp=(new,catlg) **temp** &&name disp=(new,pass)
* deleting an existing dataset - disp=(old,delete)
* catalog and uncatalog dataset - disp=(old,catlg/uncatlg)
* creating GDG generations - filename(+1), disp=(new,catlg) // MOD to append

**IEBGENER:**

* **Copying** data from one PS to another PS. **SYSUT1** (in) and **SYSUT2** (out).
* Can also **concatenate** multiple ps files. By having multiple DSN in SYSUT1.

**IEBCOPY:**

* **Copies members** from one PDS to another PDS. **COPY** INDD=dd1,OUTDD=dd2
* Con also **compress** PDS. COPY INDD=dd1,OUTDD=dd1
* You can: SELECT MEMBERS=, EXCLUDE MEMBERS=, replace by (name, ,R)

**SORT:**

* **Sort** SORT FIELDS=(starting point (sp), length, CH or ZD, and A or D), eliminate duplicates with SUM FIELDS=NONE, **Merge** MERGE FIELDS=(1,6,zd,A), **Copy** SORT FIELDS=COPY, **Split** OUTFIL, INCLUDE= and **format** datasets inrec/outrec.

**Procedure** is a collection of pretested JCL statements. There are two types: **Instream procedures** (max 15) are defined and executed within the same JCL. **Catalog procedures** are pre-defined and stored in a procedure library, and they can be called by any JCL.

* **Instream Procedure:** Defined between the PROC and PEND statements within JCL.
* **Catalog Procedure:** Referred to using the EXEC statement with the PROC name.

**Symbolic parameters** are placeholders used in procedures to allow flexibility and reusability. **Syntax:** They are substituted with actual values when the procedure is invoked. Symbolic parameters are defined with an ampersand (&) in the procedure and replaced by actual values in the EXEC statement.

**Gdg** (generation data group) is a collection of like-named datasets which are chronologically related to each other. the **two rules;** first you create a gbg base with idcams and then you create a generation with iefbr14. Current generation will always be zero.

**Syntax:**

The parameters in a GDG BASE MODE:

* **DEFINE** GDG(NAME(ID.JCLMYBASE) to be the gdg base name.
* **LIMIT**(5) is maximum 5 generations **max is 255**.
* **noempty**: when limit is max, and you rolllin a new generation then the oldest generatrion will get rolledoff which means it gets means uncataloged. **empty**: when limit is max, and you rolllin a new generation then all the existing generations will be rolledoff, only the new geneartion will be present.
* **scartch** means the uncatalog will be deleted, **noscratch** means the uncataloged generations should not be deleted.

**Referback** - referring back to a previously dd statement.

\*.STEP-NAME.DD-NAME -< REF. BETWEEN THE STEPS   
\*.DD-NAME -< REF. WITHIN A STEP

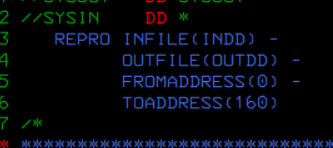
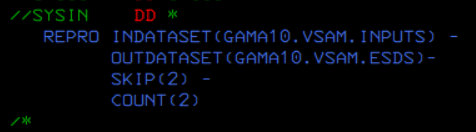
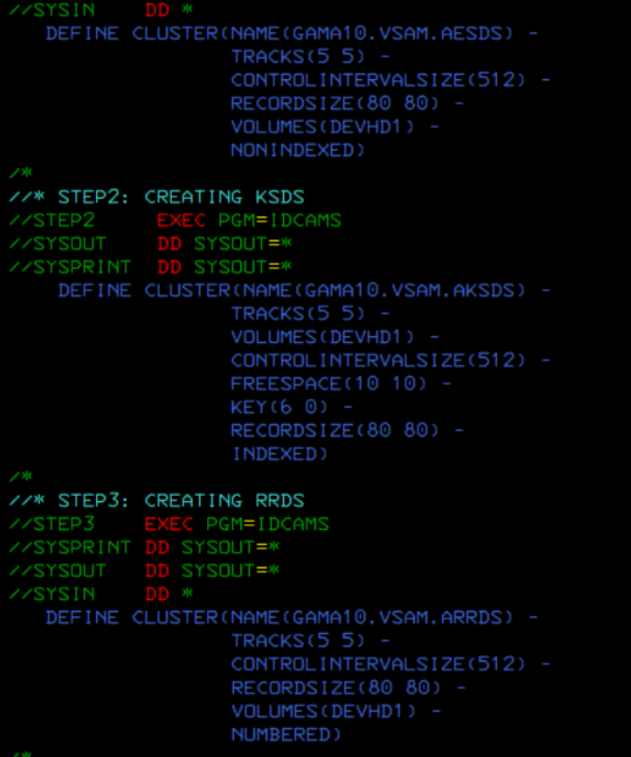
# VSAM

**VSAM Datasets** are known as clusters - **cluster** provides building block/structure = **collection of records to vsam.**

**Also Cluster** offers **components** to hold records. One of the components is data where all the data is stored. In catalog for vsam datasets you will get a **minimum of two entries** one for cluster and one for data.

**types of vsam datasets:** esds, ksds, linear ds and rrds. also only more vrrds (extensive version of rrds).

* esds - non indexed - sequential - entry sequeitanl dataset. will store inputs exactly as given - raw data - can have alternative index
* ksds - indexed - you can select the key field. Duplicate records are not allowed, keys must be unique - can have alternative index
* rrds - numbered - similar to esds but numbered, unlike esds cannot be variable length for that we can use vrrds - cannot have alternative index
* Lds - linear

Jcl utility is dedicated with vsam. Unload - fromaddress/toaddres, Load - Count , Skip

Printing default is dump; have to specify char or hex

According to cluster starting point is 0, that's why we call it an offset.

**rba** (relative byte address) is the record's starting point starting the index from 0 not 1. rba is how cluster identifies the records in it

**Alternate Index (AIX)** is used to access a KSDS (Key-Sequenced Data Set) file using a key that is different from the primary key.

Data in the cluster gets stored in the **control interval.** (four things are logical space, free space, ci, rdf) minimum for control interval is track (512) and max is cylinder (4096) Everything outside is a **control area**.

**LOAD** is when you collect from another ps. for example: every 6 hours for an atm in real life you have to use the load utility and the data will be stored in a database. load values? Values from the file you collect.

Ways to **print vsam** datasets: 1st way **spool** u can view vsam cluster. 2nd way **catalog** ‘print ids)’ code to print. 3rd way **file manager.** 4th way is with **ditto** 1-3 and enter name in ' '.

# COBOL

**Definition:** COBOL (common business oriented language) - **application language** - to create commercial applications. Cobol consists of four divisions. which are in hierarchy: identification division, environment division, data division and procedure division.

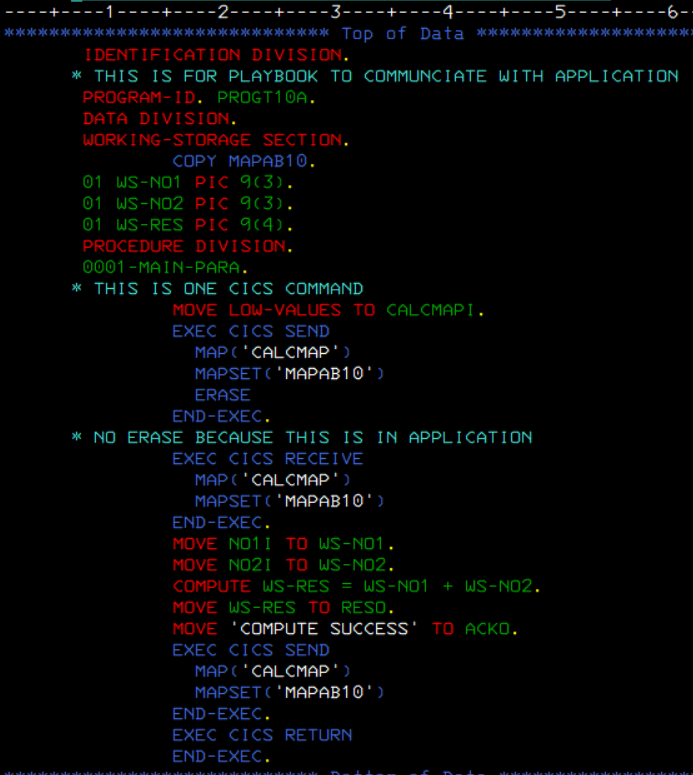
* **Identification division** - providing the basic details about the program, creator, when was it last compiled, which organisation - this is compulsory for every program.
* **environment division** - mainly for external programs - if there is no need for external resources for the program then environment division is not needed, it's optional.
* **data division** - declaration variables - all declaration is done in data / data within the program.
* **procedure division** - all your code goes here.

Cobol data division has different declarations: **file section** to build a record structure, **working-storage section** to declare the local variables, **linkage section** is for sharing variables with other files, report section and screen section.

Cobol areas: **Area A** (8-11 cols - paragraphs should start with area A) and **Area B** (12-27 all your statements and sentences should start from area B).

Three basic **cobol verbs** which help execute programs are **display, accept and stoprun** are coded on area B.

**load module** is the output for cobol in binary format. The pds which hold the load module are called load libraries.

ADD SUBTRACT MULTIPLY DIVIDE

**Img of cobol:**

# CICS

It's a subsystem like JES. **Used to execute online programs for uses to interact with.**

**Transaction** is a request and response. To initiate a transaction, we have to provide a transaction id 4 bytes long).

**Terminal** understands transaction id, **CICS** understands commands.

Terminal is where a customer does tasks.

**Two types of transaction ids:** CICS Supplied and User Defined. **difference is:** CICS always start with C and user defined ids cannot start with C.

**online program created** in tso/ispf, compile in jcl - **how to compile?** with sub, execute in CICS, **how to execute?** you define transaction id (which is user define) first then the application program and install / update. **Then execution** in terminal.

**CICS commands:** 1. Send, 2. Receive, 3. Return (to stop / end)

cus: provide transaction id then enter -> you: send empty map to the terminal / what else can you send? has to be empty -> cus: fill pnr number (populating the map) and click submit -> you: recieve the populated map from customer (the pnr number) -> you: file related commands and collect information from the databases so send - file/data processing with filled info -> and finally after sending the filled info - the response is over return.

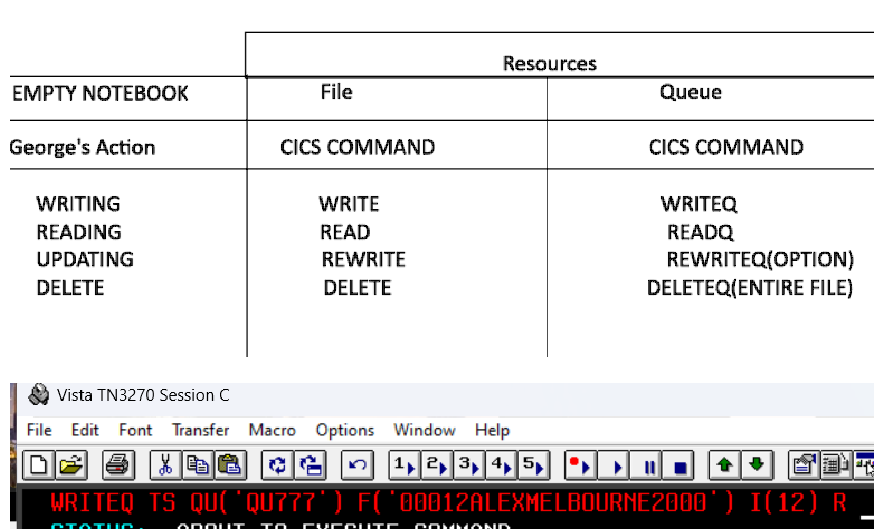
**ceda** - can do administrator activities - examples: **Define and Install** (can only define here)

**cemt** - execute master terminal - is a level 2 transaction id - for operators - you can use **inquire** check everything that is installed - can also install with SET

**ceci** - is for doing prototypes or / testing without the application program. For example printing a map you can use the syntax: SEND MAP(name) MAPSET(name). Can also use other cics command such as write, read, rewrite, delete

## Queue

is a collection of items. Two types of queues - ts (temporary storage) queues and td (temporary data) queues.

R is for rewrite its optional

CEBR - EXECUTION BROWSE - to check queue

## BSM

**Map** is output interface it’s for the users. **Two types of**: Physical map consists of physical fields and protected fields. **Symbolic map** consists of symbolic fields and unprotected fields.

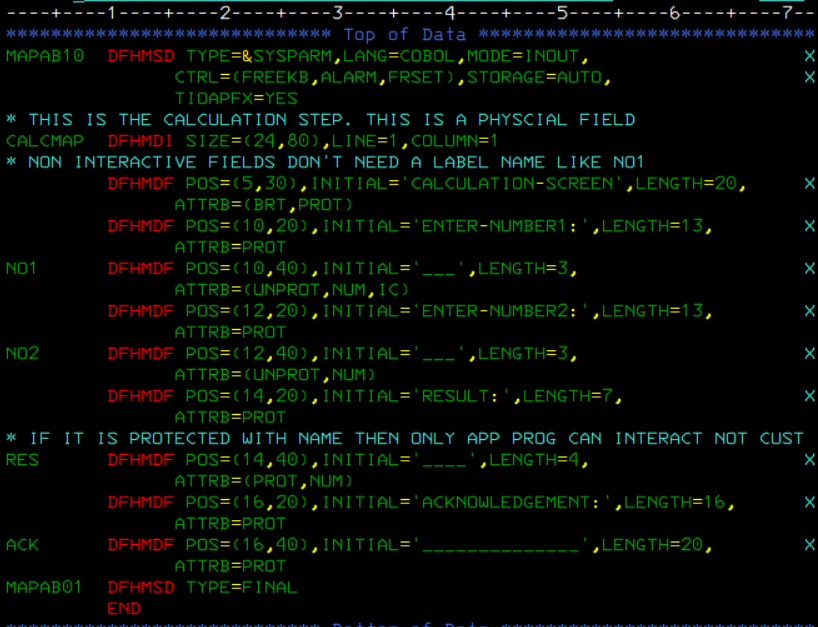
Mapset is a collection of maps, a map is a collection of fields.

**BMS** (basic mapping sort) is made up of marcos and assembly language - there are three macros.

* **DFHMSD** is for mapset definition.
* **DFHMDI** is for map definition.
* **DFHMDF** is for field definition.

**COPYBOOK** is a pds member = SYMBOLC MAP - it's made after you submit a mapset - you can see all the input and output variables

when you request you receive two numbers from the symbolic playbook (**input flow**). One of the two numbers will be from no1i and the next will be found in no2i. to deliver you need to use no1o (**output flow**).

**Img of BMS:**

# DB2

is a database management system that provides a comprehensive environment for managing and manipulating data. Understanding the various SQL statements and their corresponding keywords is crucial for effectively working with DB2.

### **SQL Statements (4 Categories)**

**1. Data Definition Language (DDL) Statements**

DDL statements define and manage the structure of database objects. The primary DDL statements are:

* **CREATE**: Creates a new table, view, index, or other database objects.
* **ALTER**: Modifies an existing database object, such as a table.
* **RENAME**: Renames a database object.
* **DROP**: Deletes a database object.

**2. Data Manipulation Language (DML) Statements**

DML statements manipulate the data within the database. The primary DML statements are:

* **INSERT**: Adds new rows to a table.
* **SELECT**: Retrieves data from one or more tables.
* **UPDATE**: Modifies existing data within a table.
* **DELETE**: Removes rows from a table.

**3. Data Control Language (DCL) Statements**

DCL statements control access to data within the database. The primary DCL statements are:

* **GRANT**: Gives permissions to users to perform certain actions.
* **REVOKE**: Removes permissions from users.

**4. Transaction Control Language (TCL) Statements**

TCL statements manage the changes made by DML statements. The primary TCL statements are:

* **COMMIT**: Saves all changes made during the current transaction.
* **ROLLBACK**: Undoes all changes made during the current transaction.

### **Keywords**

* **PRIMARY KEY**: Uniquely identifies each record in a table.
* **FOREIGN KEY**: Establishes a relationship between two tables.
* **UNIQUE KEY**: Ensures all values in a column are unique.

### **Functions**

#### **Arithmetic Functions:**

Perform basic arithmetic operations:

* +, -, \*, /

#### **Scalar and Aggregate Functions:**

**Scalar Functions**: Operate on each row and can be used in the WHERE clause.

* **CHAR**
* **BLOB**
* **BINARY**
* **BIGINT**
* **SMALLINT**
* **DATE**
* **DECIMAL**
* **DEGREES**
* **TIME**
* **HEX**

**String Functions**:

* **CONCAT**
* **STRIP**
* **TRIM**
* **SUBSTR**
* **STRPOS**
* **LOWER**
* **UPPER**

**Math Functions**:

* **ABS**
* **FLOAT**
* **FLOOR**
* **SQRT**
* **MIN**
* **MAX**

**Date/Time Functions**:

* **DAY**
* **MONTH**
* **DAYOFWEEK**
* **DAYS**
* **HOUR**

**Array Functions**: (Varies with specific DB2 implementations)

**Aggregate Functions**: Operate on multiple values (rows) and can only be used in the SELECT clause.

* **COUNT**
* **AVG**
* **MAX**
* **MIN**
* **SUM**
* **PERCENTILE\_**\*
* **VARIANCE**

### **Querying from Multiple Tables**

**Set Operations**:

* **UNION**: Combines the results of two or more queries.
* **INTERSECT**: Returns the common results of two queries.

**JOINs**:

* **INNER JOIN**: Returns rows that have matching values in both tables.
* **LEFT JOIN**: Returns all rows from the left table, and the matched rows from the right table.
* **RIGHT JOIN**: Returns all rows from the right table, and the matched rows from the left table.
* **OUTER JOIN**: Returns all rows when there is a match in one of the tables.
* **CROSS JOIN**: Returns the Cartesian product of the two tables.